BROWN

Track reconstruction

T. Speer

28 April 2010

Outline

- The Combinatorial Track Filter
  - The Kalman Filter
  - The combinatorial Kalman Filter
- Robust algorithms
- Results from CRAFT and the first collisions

Least square estimators

- The Kalman Filter is mathematically equivalent to a global Least-Square minimization (LS)
- LS estimators optimal when
  - model is linear
  - random noise Gaussian
- If the model is linear and random noise is Gaussian:
  - LS estimators are unbiased and have minimum variance
  - Residuals and pulls of estimated quantities are also Gaussian
  - Objective function obeys a χ^2 distribution
- For non-linear models or non-Gaussian noise, LS still the optimal linear estimators

The Kalman Filter

- Method to estimate the states of a dynamic system
  - Used in time-series analysis, signal processing
- Recursive
  - Track parameters estimated from a set of reconstructed hits
  - One hit added at a time, updating the state
  - No large matrices need to be inverted
- Track is described as discrete dynamic system
  - System equation:

$$x_k = f_k(x_{k-1}) + \delta_k$$

  - Evolution of the track in the tracking detector
  - x_k : State of the track on layer k
  - f_k : track model, between layer (k-1) and k
  - δ_k : Process noise, between layer (k-1) and k (multiple scattering)

$$< \delta_k >= 0, \quad Cov[\delta_k] = Q_k$$

The Kalman Filter

- Measurement equation:
  - dependence of the observations on the local state

$$m_k = h_k(x_k) + \epsilon_k$$

  - m_k : measurement in layer k
  - h_k : measurement model
  - ε_k : measurement error
  - V_k : Covariance of the measurement

$$< \epsilon_k >= 0, \;\; Cov[\epsilon_k] = V_k = G_k^{-1}$$

The Kalman Filter

- Measurement equation:
  - dependence of the observations on the local state

$$m_k = h_k(x_k) + \epsilon_k$$

  - m_k : measurement in layer k
  - h_k : measurement model
  - ε_k : measurement error
  - V_k : Covariance of the measurement

$$< \epsilon_k >= 0, \;\; Cov[\epsilon_k] = V_k = G_k^{-1}$$

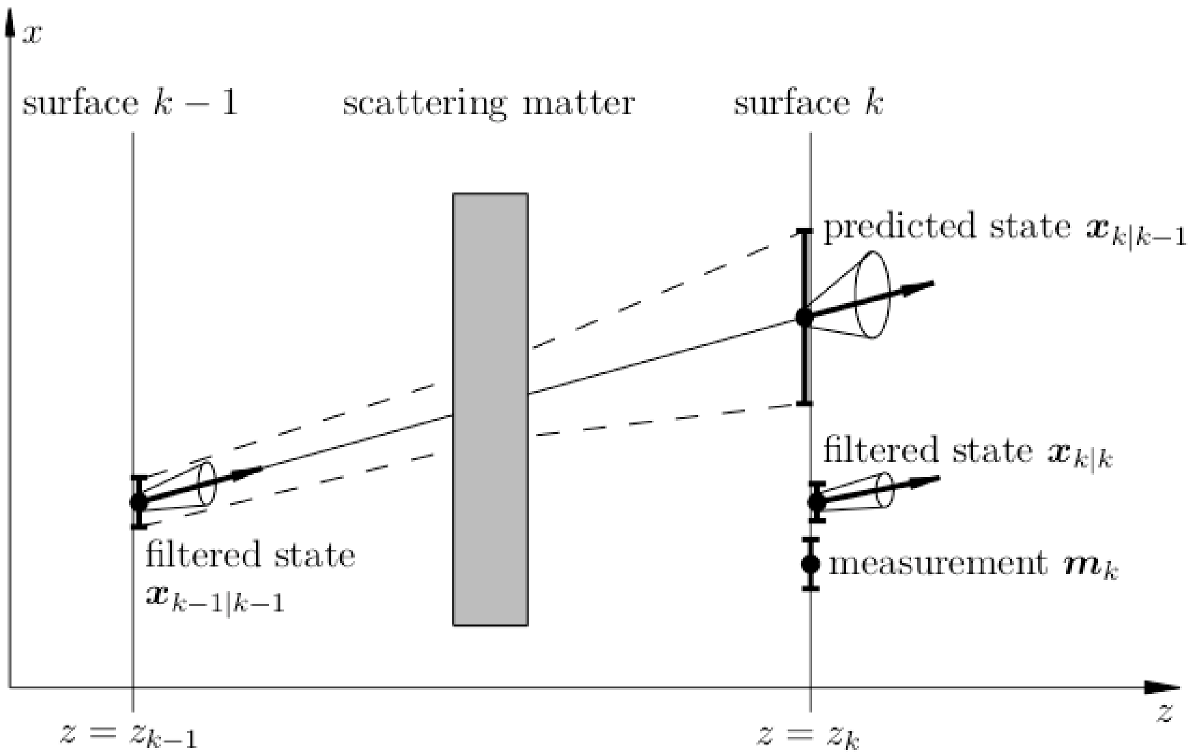
- If track model is not linear, first-order Taylor expansion used:

$$f_k(x_{k-1}) \approx f_k(x_e) + F_k(x_{k-1} - x_e) = F_k x_{k-1} + c_k$$

$$h_k(x_{k-1}) \approx h_k(x_e) + H_k(x_{k-1} - x_e) = H_k x_{k-1} + d_k$$

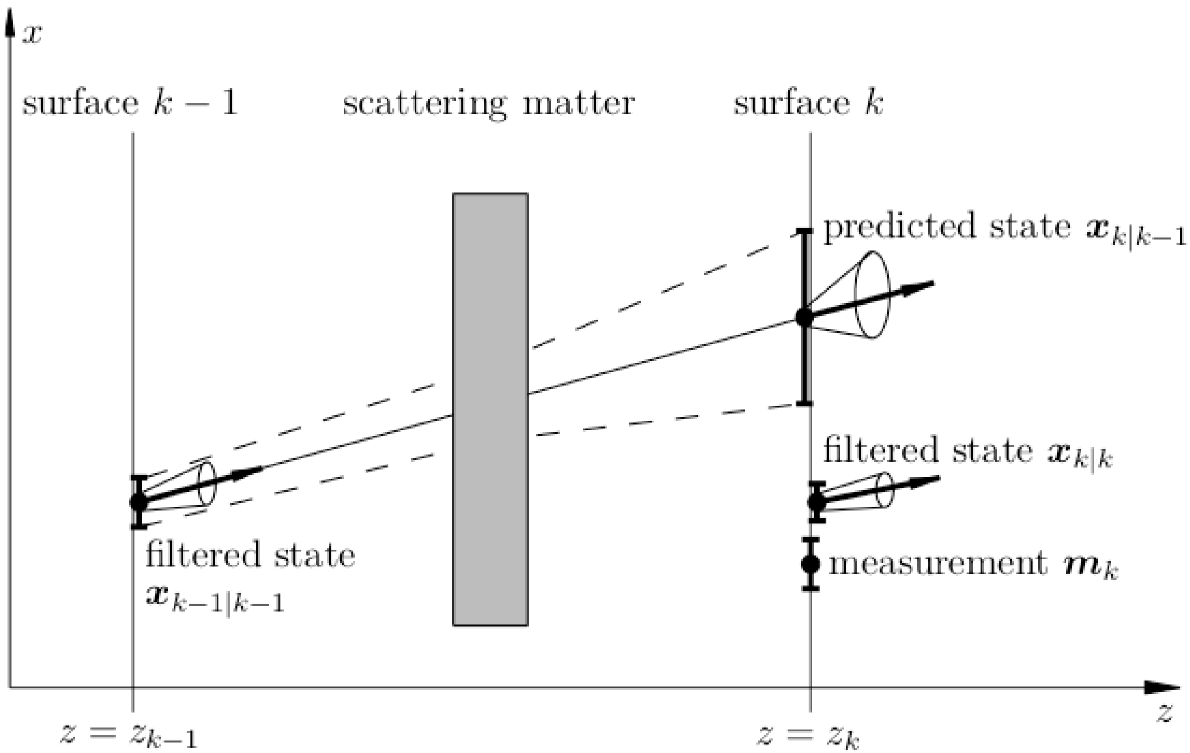
$$F_k = [\partial f_k / \partial x_k]_e \;\;, \;\; H_k = [\partial h_k / \partial x_k]_e$$

The Kalman Filter



- Recursive procedure:
  - Prediction
  - Filtering
  - Smoothing

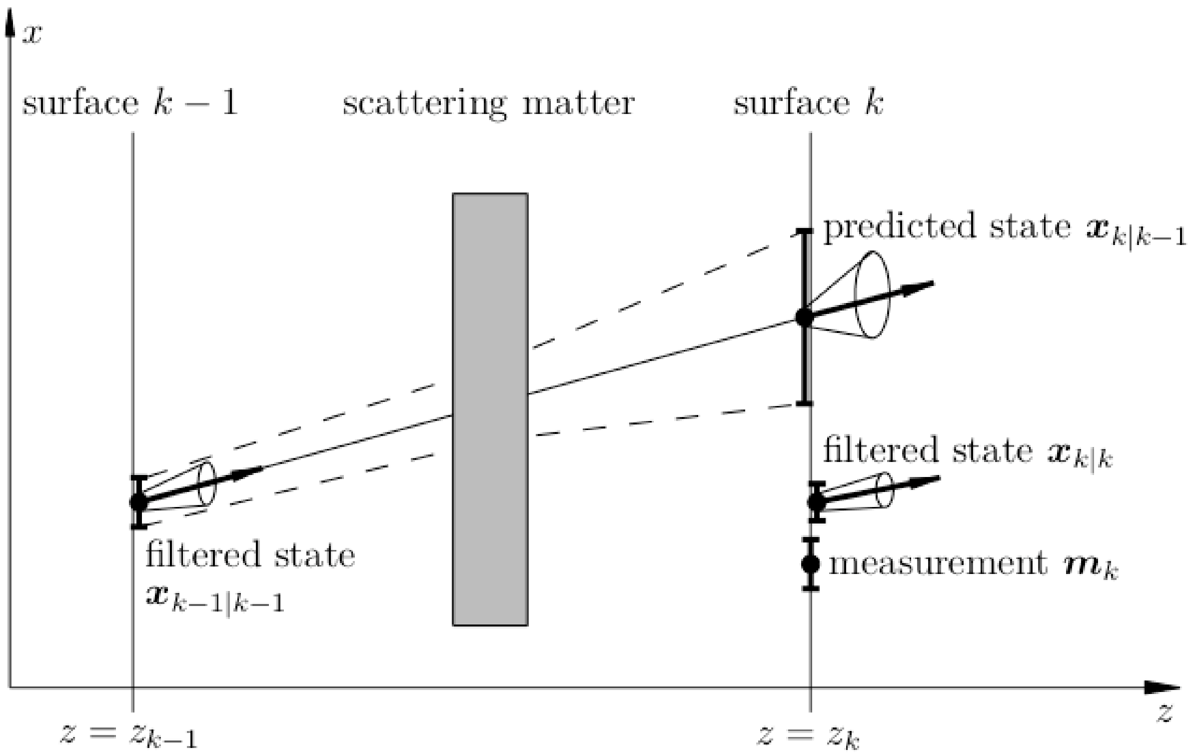
The Kalman Filter



- Prediction: Trajectories are extrapolated from a layer to next layer, accounting for multiple scattering and energy loss
  - Prediction of state vector $x_{k|k-1}$, based on measurements $Y_{k-1} = \{y_1, \ldots, y_{k-1}\}$:

$$x_{k|k-1} = F_k x_{k-1|k-1} \quad , \quad C_{k|k-1} = F_k C_{k-1|k-1} F_k^T + Q_k$$

The Kalman Filter



- Filtering: On the new layer, new trajectories are constructed, with updated parameters (and errors) for each compatible hit in the layer.
  - Weighted mean of prediction and observation
  - $x_{k|k}$, based on measurements $Y_k = \{y_1, ..., y_k\}$:

$$x_{k|k} = C_{k|k}[C_{k|k-1}^{-1} x_{k|k-1} + H_k^T G_k m_k] \quad , \quad C_{k,k} = [C_{k,k-1}^{-1} + H_k^T G_k H_k]^{-1}$$

The Kalman Filter

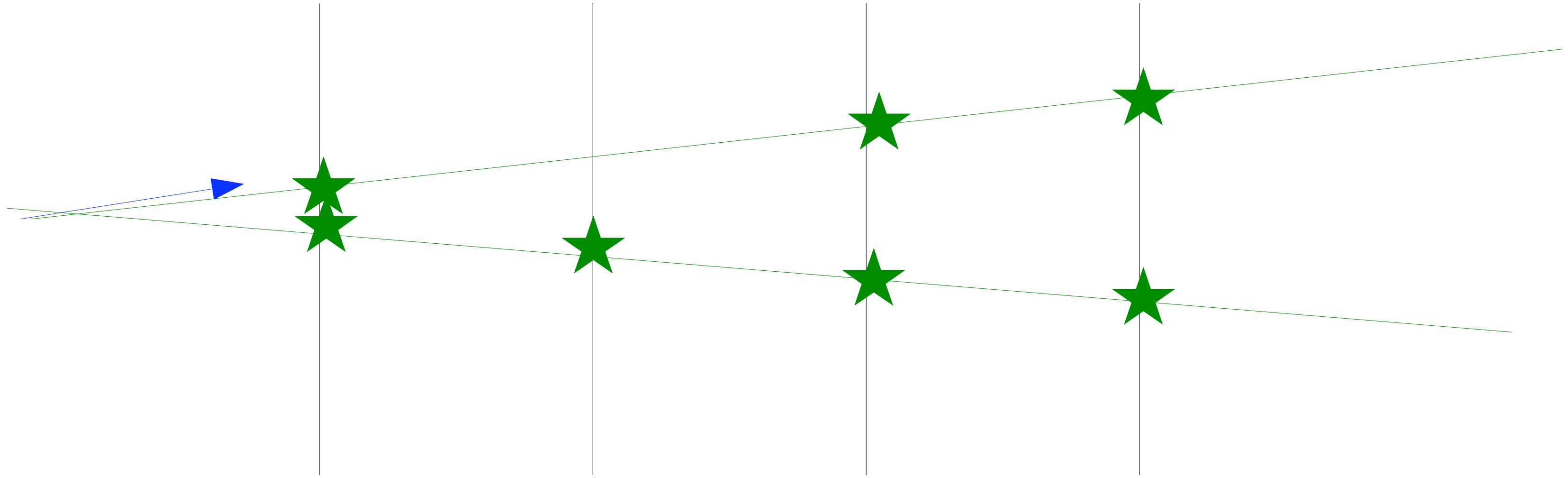
- Prediction: Trajectories are extrapolated from a layer to next layer, accounting for multiple scattering and energy loss
- Filtering: On the new layer, new trajectories are constructed, with updated parameters (and errors) for each compatible hit in the layer
- Smoothing: final fit of trajectories
  - Obtain optimal estimates at every measurement point along the track
  - In addition to providing tracks accurate at both ends this procedure provides more accurate rejection of outliers
  - Combination of forward and backward filters by a weighted mean

The Combinatorial Kalman Filter

- Integrate track fitting and pattern recognition
- Build track(s) from an initial trajectory (*seed*)
  - Combinatorial exploration of all possibilities
  - Build all candidates in parallel to avoid bias

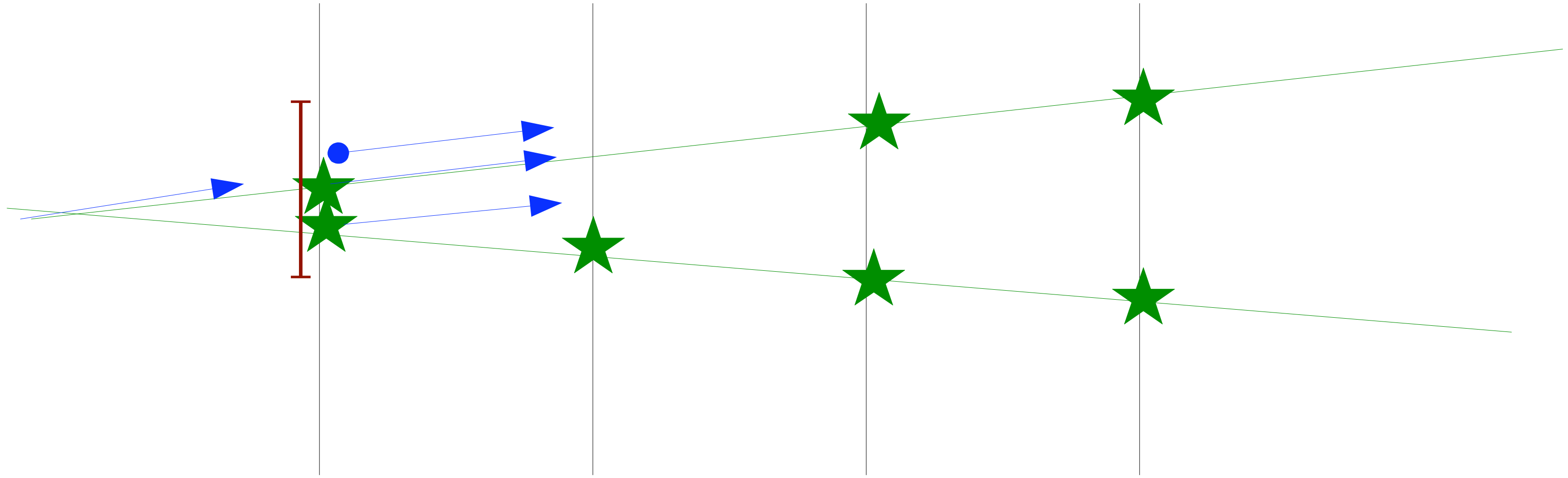
The Combinatorial Kalman Filter

- Starting from a *seed*:
  - Extrapolated trajectory from layer to next layer, accounting for multiple scattering and energy loss (KF)



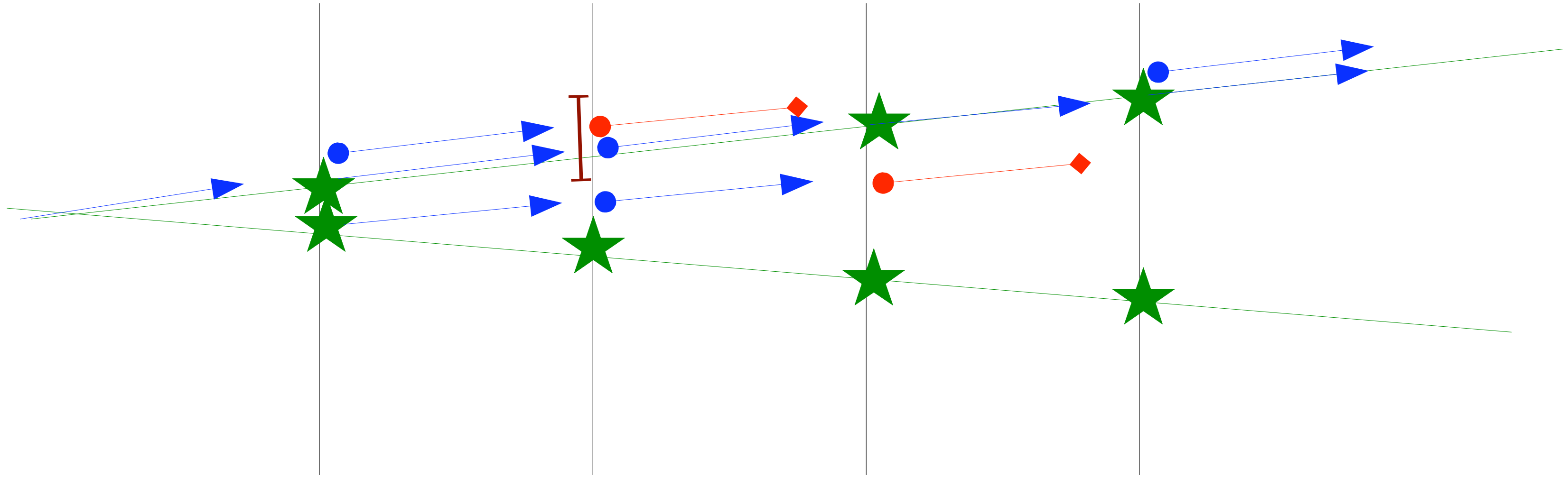
The Combinatorial Kalman Filter

- Starting from a *seed*:
  - Extrapolated trajectory from layer to next layer, accounting for multiple scattering and energy loss (KF)
  - On the new layer, constructed new trajectories, with updated parameters (and errors) for each compatible hit in the layer.
    - One additional trajectory is added without new measurement (invalid hit/fake hit



The Combinatorial Kalman Filter

- Starting from a *seed*:
  - Extrapolated trajectory from layer to next layer, accounting for multiple scattering and energy loss (KF)
  - On the new layer, constructed new trajectories, with updated parameters (and errors) for each compatible hit in the layer.
    - One additional trajectory is added without new measurement (invalid hit/fake hit)
  - All trajectories are grown to the next layer in parallel
  - The number of trajectories to grow is limited according to their χ^2 and the number of missing hits



The Combinatorial Kalman Filter

- Starting from a *seed*:
  - Extrapolated trajectory from layer to next layer, accounting for multiple scattering and energy loss (KF)
  - On the new layer, constructed new trajectories, with updated parameters (and errors) for each compatible hit in the layer.
    - One additional trajectory is added without new measurement (invalid hit/fake hit)
  - All trajectories are grown to the next layer in parallel
  - The number of trajectories to grow is limited according to their χ^2 and the number of missing hits
- Since trajectories are grown in parallel, large number of duplicate and overlapping trajectories
  - Final collection has to be cleaned to remove duplicates

The Combinatorial Track Finder

- Multi-step iterative Combinatorial Kalman Filter
- Decomposed in modular, independent, components:
    - Local reconstruction: hit reconstruction

The Combinatorial Track Finder

- Multi-step iterative Combinatorial Kalman Filter
- Decomposed in modular, independent, components:
  - Local reconstruction: hit reconstruction
  - Seeds generation

The Combinatorial Track Finder

- Multi-step iterative Combinatorial Kalman Filter
- Decomposed in modular, independent, components:
  - Local reconstruction: hit reconstruction
  - Seeds generation
  - Trajectory building: construction of trajectories for a given seed
    - Trajectory Cleaning: arbitration, duplicate tracks removed, based on number of shared hits and χ^2

The Combinatorial Track Finder

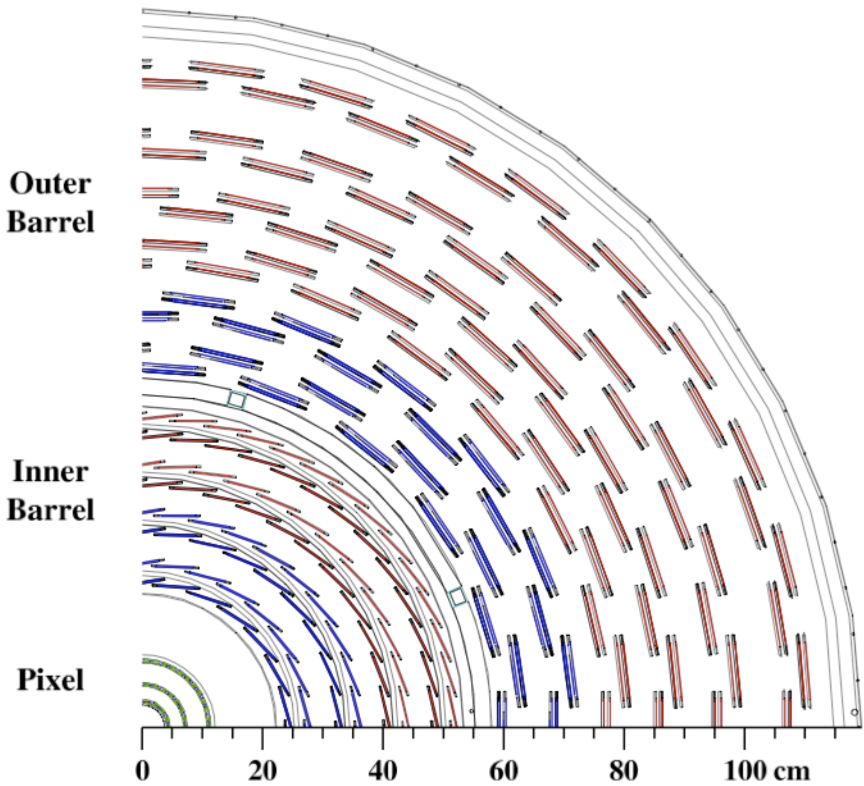
- Multi-step iterative Combinatorial Kalman Filter
- Decomposed in modular, independent, components:
  - Local reconstruction: hit reconstruction
  - Seeds generation
  - Trajectory building: construction of trajectories for a given seed
    - Trajectory Cleaning: arbitration, duplicate tracks removed, based on number of shared hits and χ^2
  - Trajectory Smoothing: final fit of trajectories
    - Outlier rejection: reject hits due noise, δ-rays, nearby tracks
    - Rejection more efficient at this stage since final fit provides optimal estimates at every measurement point along the track
    - Rejection based on χ^2 of the smoothed residual, pixel cluster probability
    - Recursive procedure: remove largest residual above threshold and refit

The Combinatorial Track Finder

- Multi-step iterative Combinatorial Kalman Filter
- Decomposed in modular, independent, components:
  - Local reconstruction: hit reconstruction
  - Seeds generation
  - Trajectory building: construction of trajectories for a given seed
    - Trajectory Cleaning: arbitration, duplicate tracks removed, based on number of shared hits and χ^2
  - Trajectory Smoothing: final fit of trajectories
    - Outlier rejection: reject hits due noise, δ-rays, nearby tracks
    - Rejection more efficient at this stage since final fit provides optimal estimates at every measurement point along the track
    - Rejection based on χ^2 of the smoothed residual, pixel cluster probability
    - Recursive procedure: remove largest residual above threshold and refit
  - Track selection:
    - Filter tracks that are likely fakes
    - Flag the expected expected of the tracks
    - Based on normalized χ^2, longitudinal and transverse impact parameters and significance

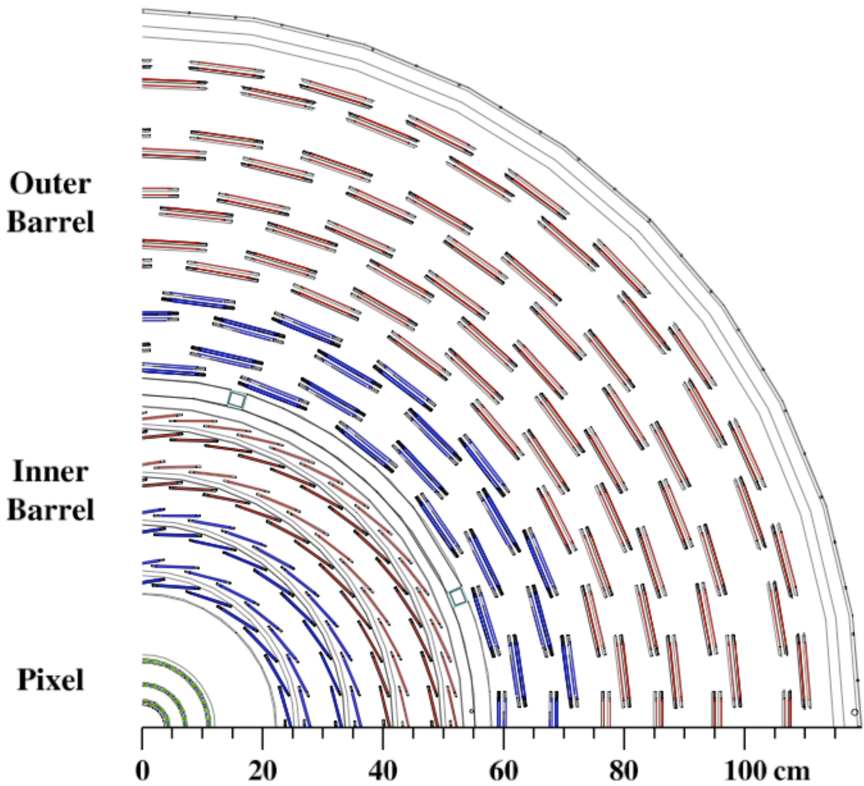
The CKF at CMS

- Outside-in tracking:
  - muon reconstruction: seeds in the outer layers based on muon-chamber seeds
  - electrons from γ conversions: seeds in the outer layers based on ECAL clusters



The CKF at CMS

- Outside-in tracking:
  - muon reconstruction: seeds in the outer layers based on muon-chamber seeds
  - electrons from γ conversions: seeds in the outer layers based on ECAL clusters
- Inside-out tracking:
  - Start in the first Pixel layers, grow tracks layer by layer to the outer layer of the SST
  - Favour tracks with pixel hits:
    - High precision
    - Important for further applications (vertex reconstruction, b-tagging)
    - Fine granularity: low occupancy, high purity
    - Reasonable number of seeds, with good quality
    - Hadrons: nuclear interactions in the tracker, may not reach the outer layers
    - Electrons loose energy because of bremsstrahlung radiation



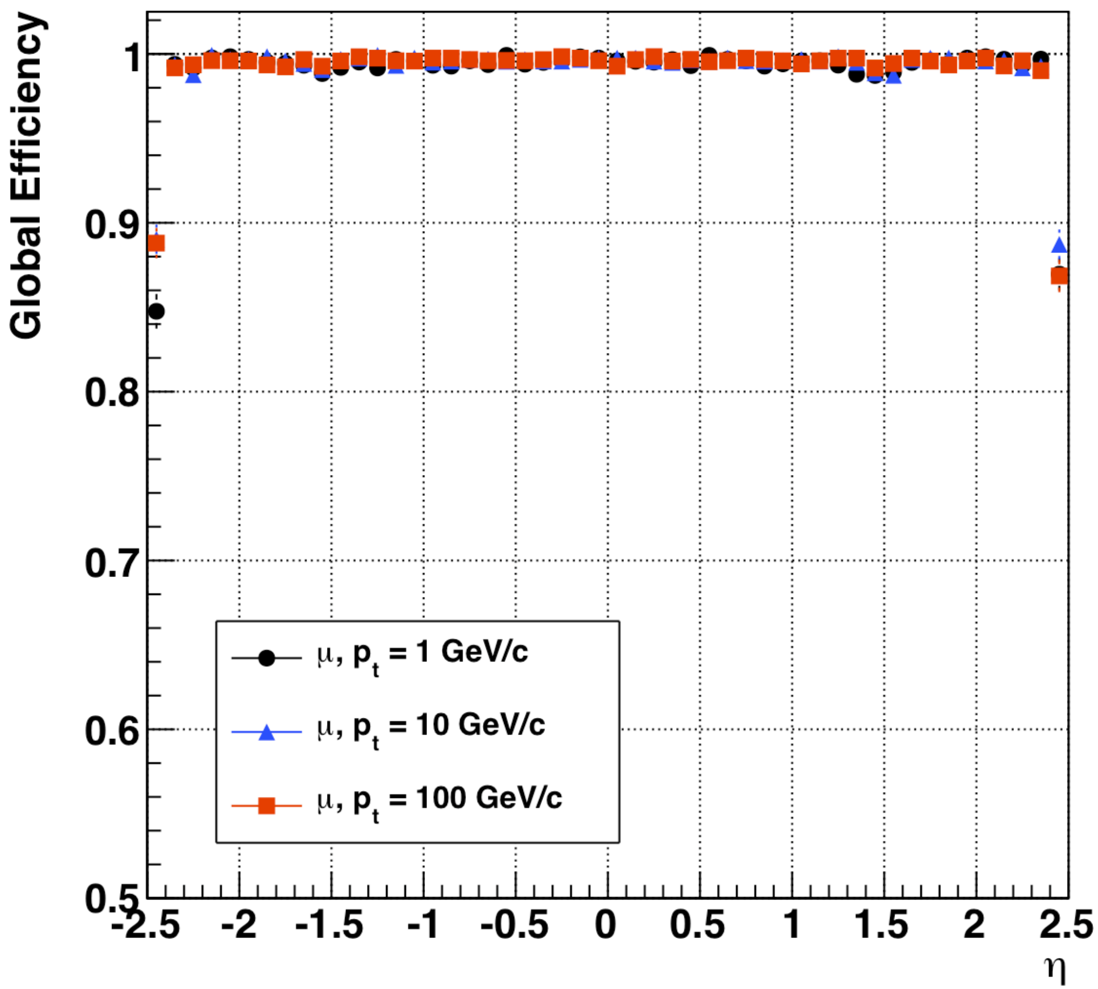
The Combinatorial Track Finder

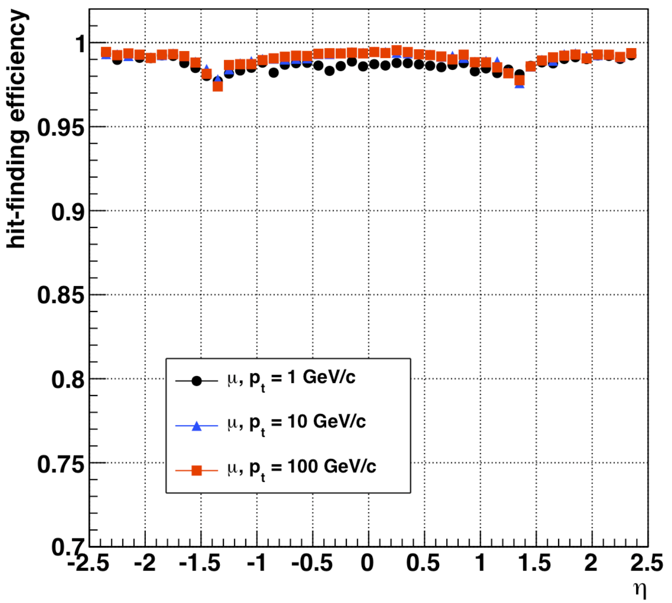
- Six-step iterative tracking sequence, with different seed types

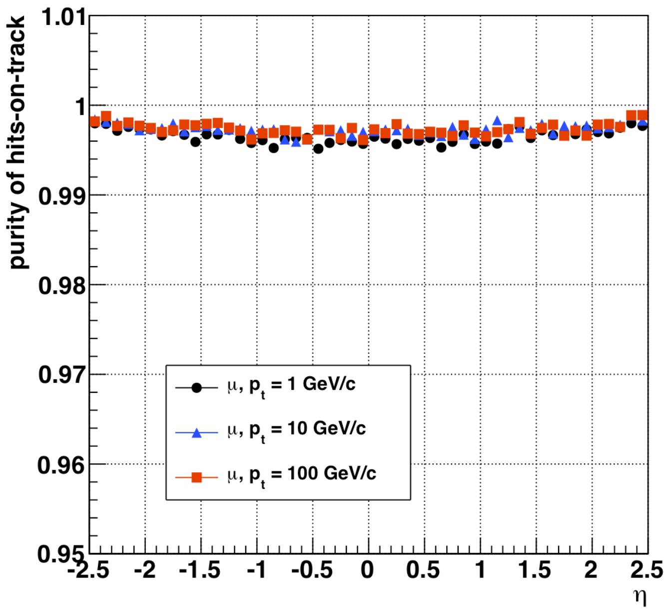
Iteration	Seeding Layers	min p_T (GeV/c)	max d_{xy} (cm)
0	pixel triplets	0.8	0.2
1	pixel pairs	0.9	0.2
2	pixel triplets	0.075	0.2
3	pixel pairs	0.35	1.2
4	TIB/TID pairs	0.5	2.0
5	TOB/TEC pairs	0.8	5.0

- After each iteration, remove hits unambiguously assigned to tracks
  - Next iteration will only use remaining hits
  - Allows to reduce the p_T threshold, Beam Spot compatibly
    - Recover tracks from V0, conversions, or for which the pixel detector was not fully efficient
- In every iteration, run full CFK-sequence on available hits
  - Different (optimised) parameters used for each iterations

Track reconstruction for single tracks: muons, p_T = 1, 10, 100 GeV/c in Monte Carlo

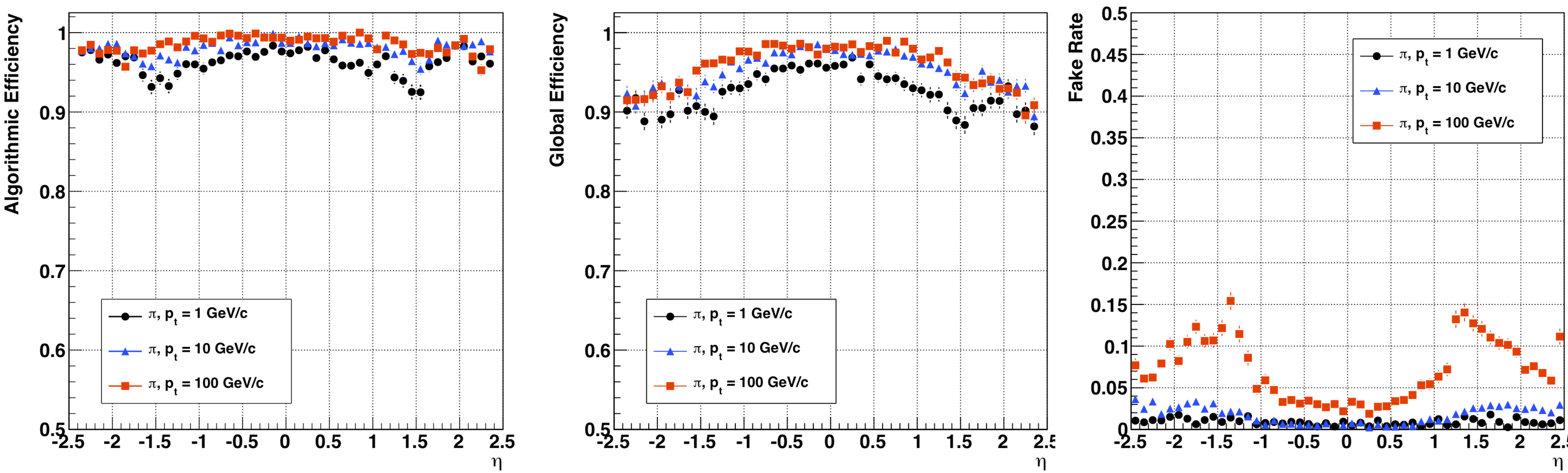






MC Track reconstruction for single tracks: pion, p_T = 1, 10, 100 GeV/*c*

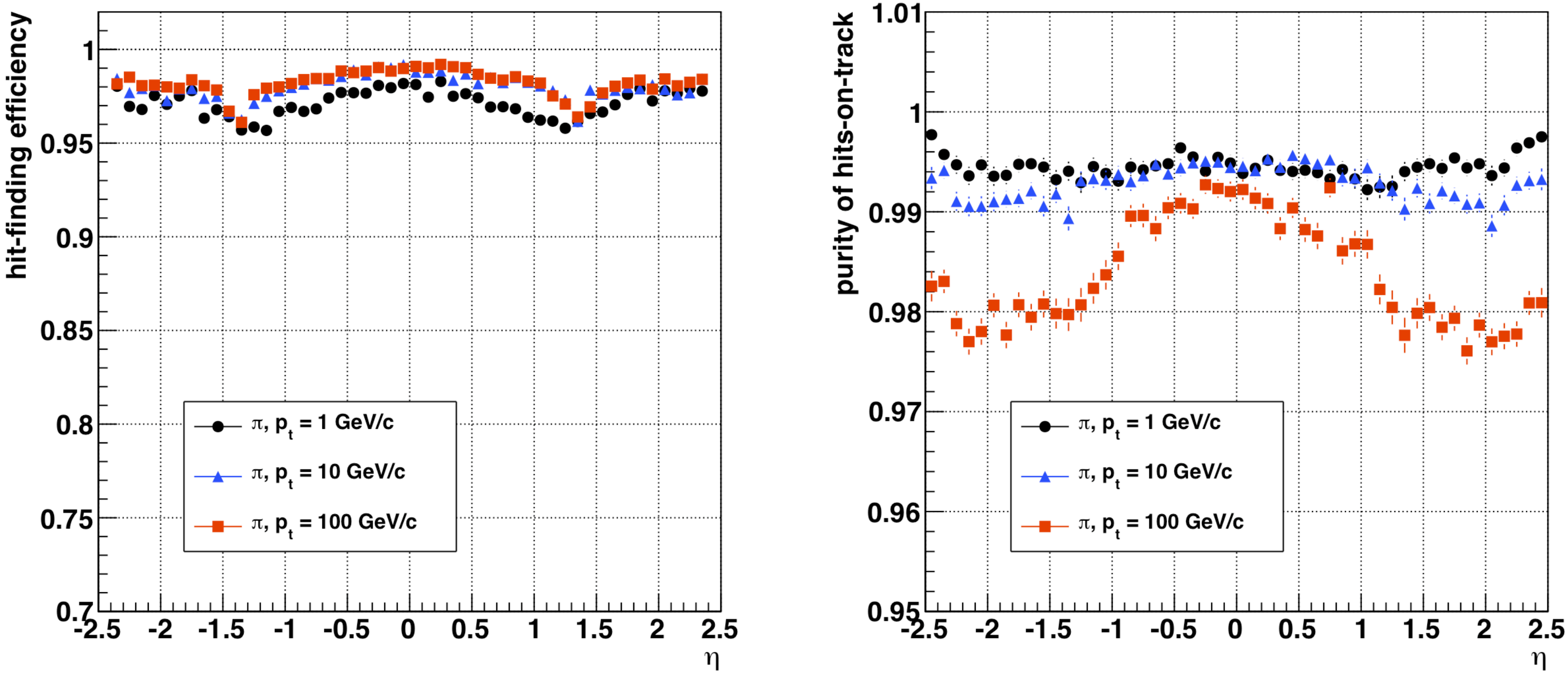
- For pions: lower efficiency due to nuclear interactions in the tracker
- Algorithmic efficiency: for simulated tracks with at least 3 hits



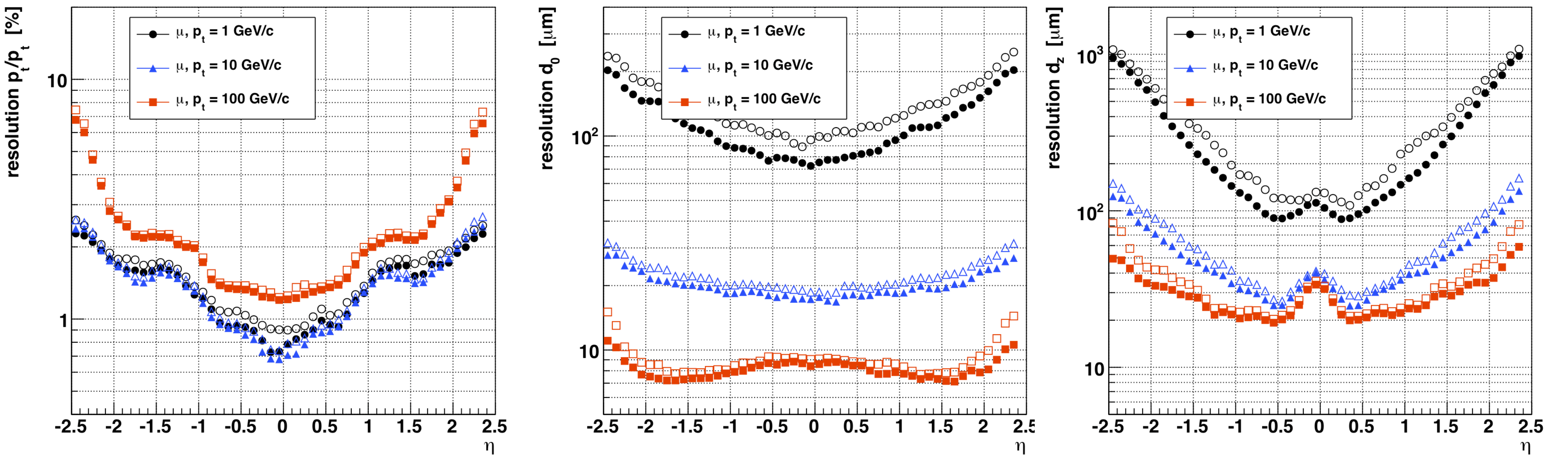
- Higher fake rate with higher energy: merging of primary and secondary particles:
  - Higher number of secondary particles
  - Smaller angle between the tracks, and smaller variation between the curvature of the primary and secondary particles.

Track reconstruction

MC Track reconstruction for single tracks: pion, p_T = 1, 10, 100 GeV/*c*



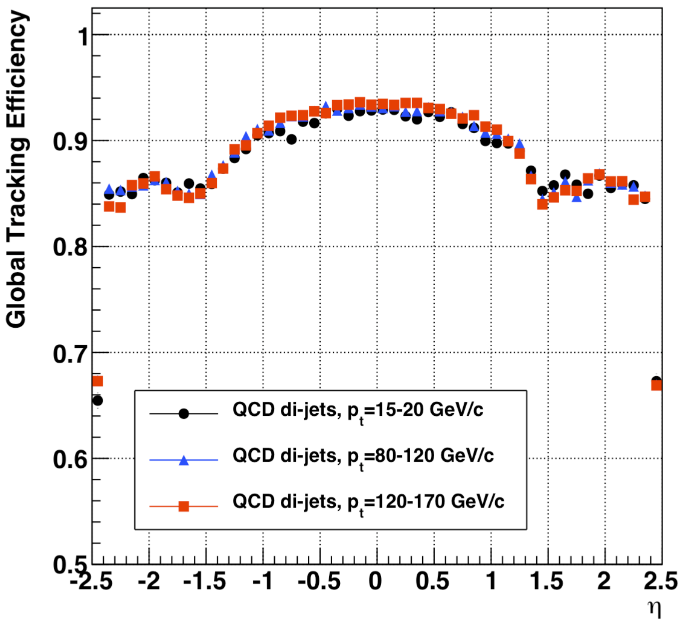
MC Track parameter resolution, single muons

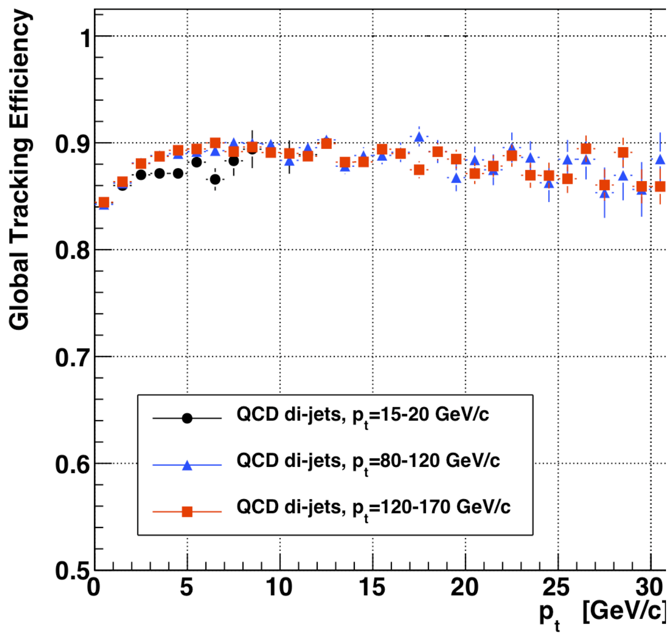


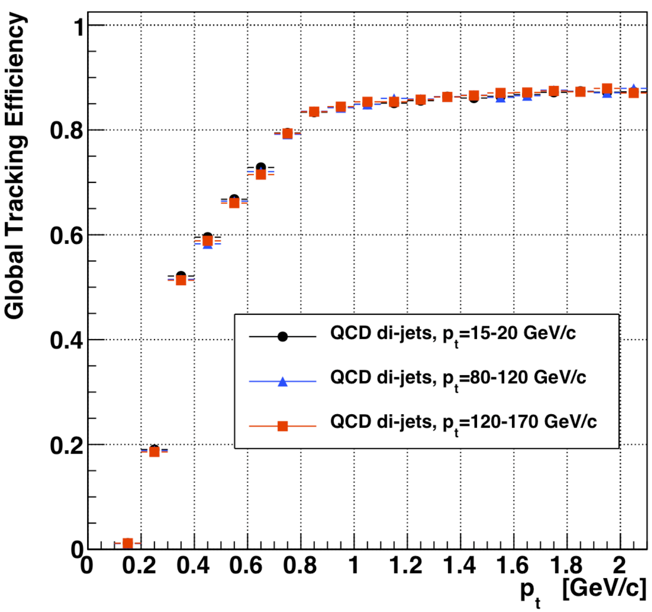
- p_T resolution: Dominated by the lever-arm
- Impact parameter resolution:
  - High momentum: Dominated by the resolution of the hits in the pixel
  - Low momentum degradation due to multiple scattering

MC Track reconstruction for dijet events

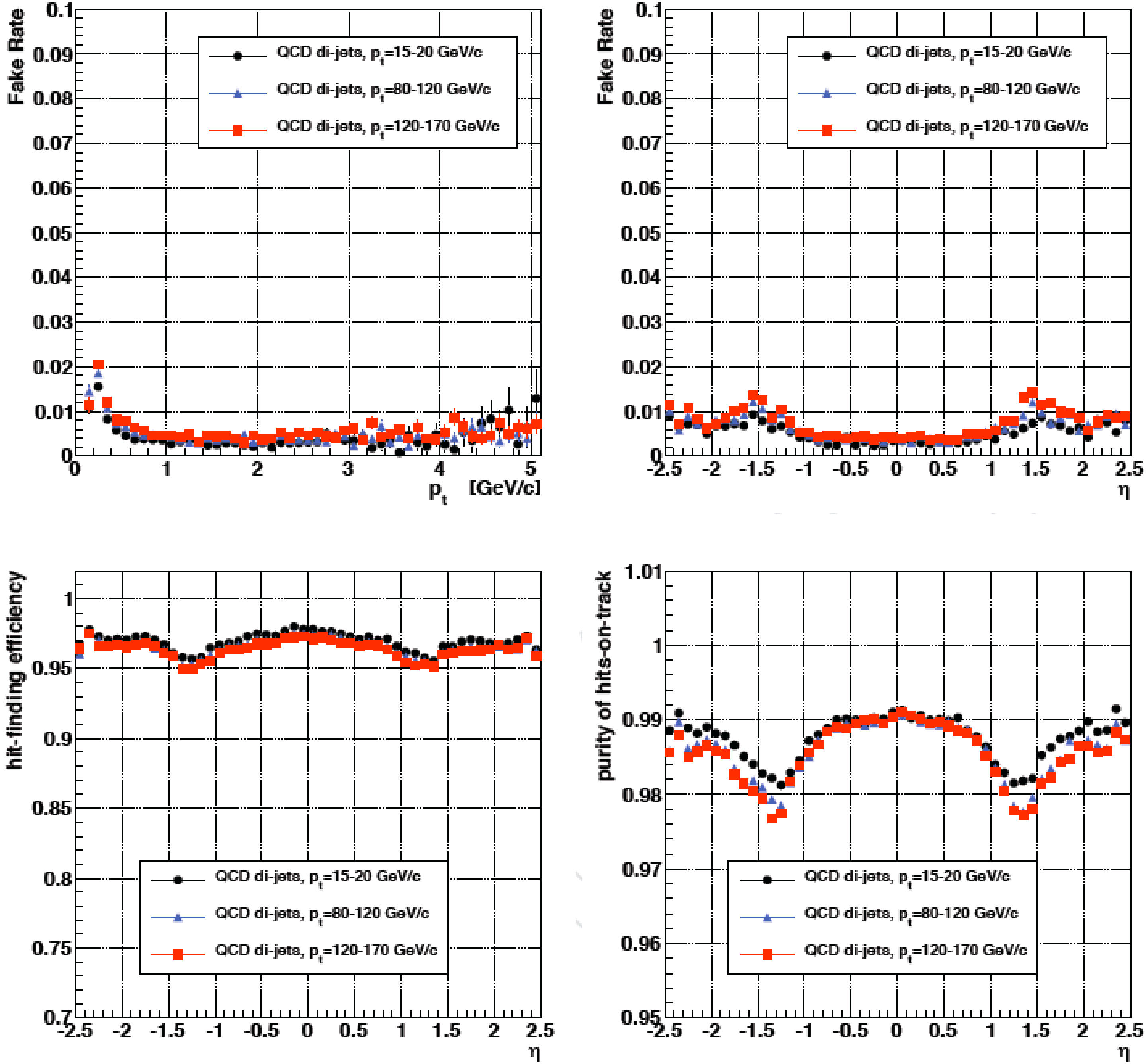
- includes underlying event, no Pile-Up







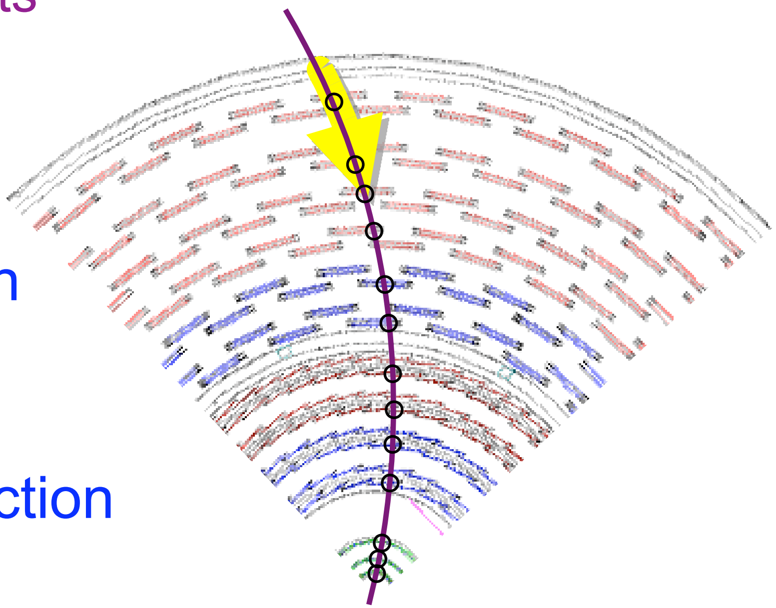
Track reconstruction



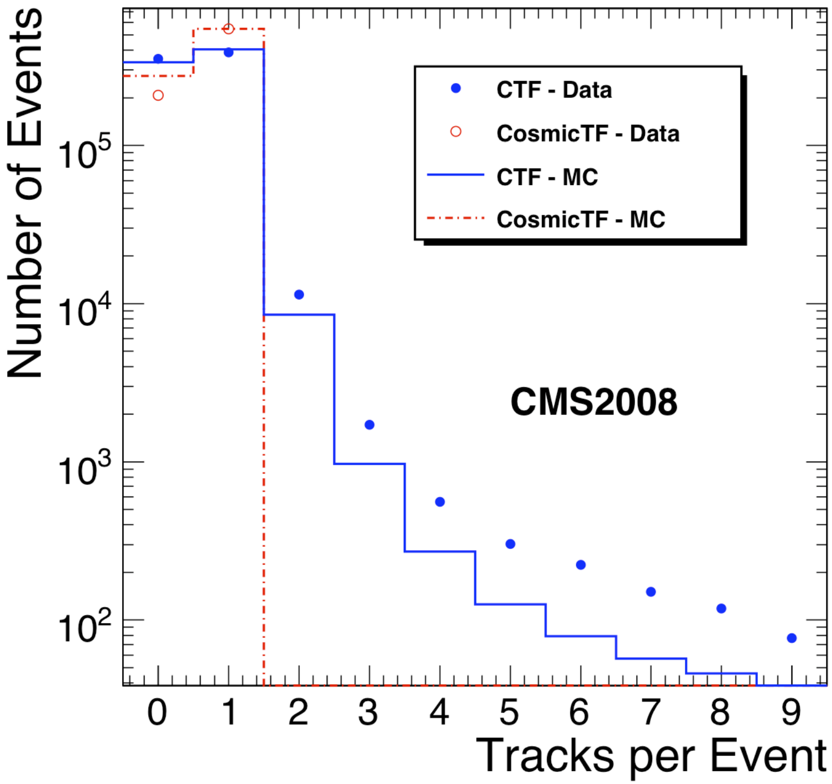
Track reconstruction in Cosmic Runs

In CRAFT runs, two algorithms used to reconstructs tracks:

- CTF: Default tracking algorithm modified for cosmic reconstruction:
  - Seeding in outer TOB and TEC layers
  - Outside-in pattern recognition with loose cuts
  - Track is reconstructed in the whole tracker
  - Reconstruction both downwards and upwards
- CosmicTF: dedicated cosmic reconstruction algorithm: 1 track/evt.
- Results from the algorithms are used as cross-check and to debug the reconstruction



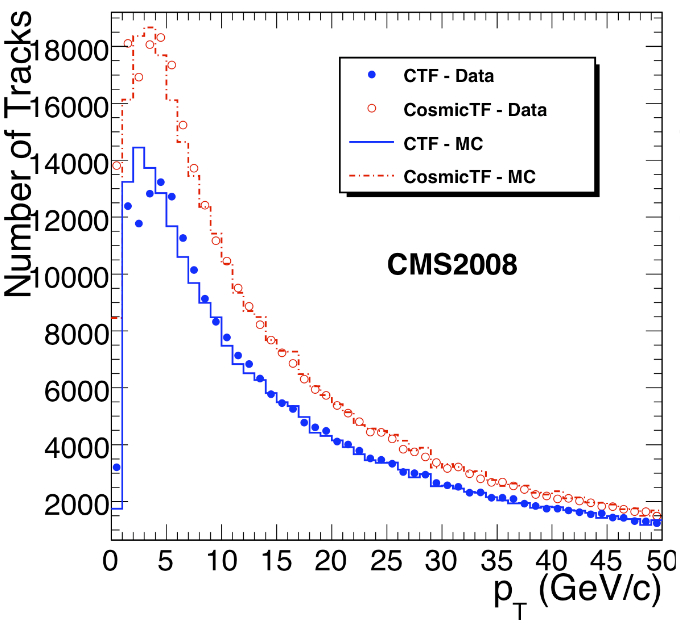
- Number of tracks per event

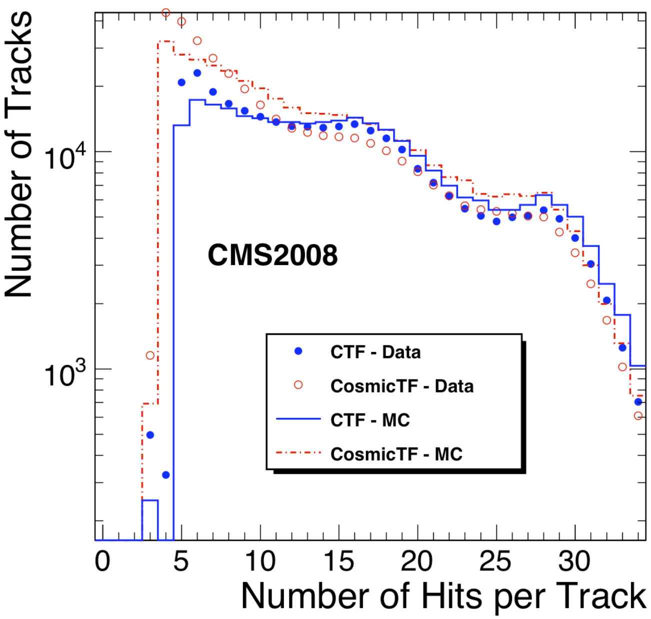


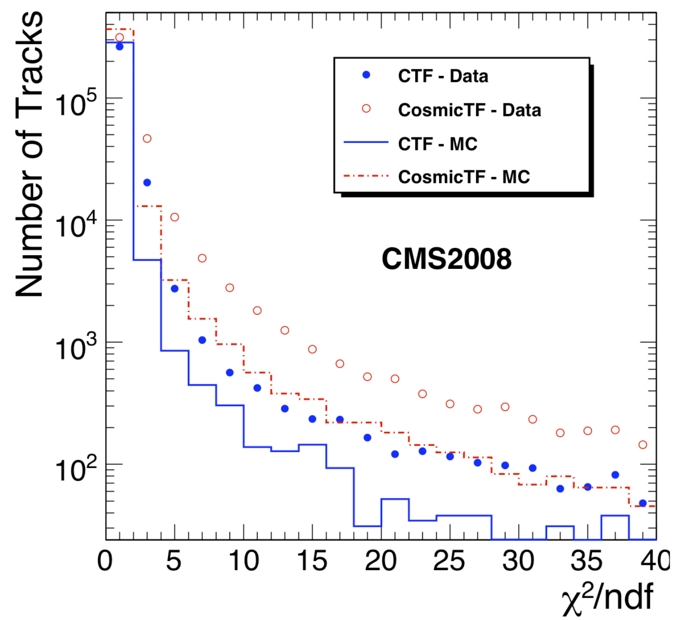
- Reconstruction of showers not optimal
- CTF not been optimised to reconstruct cosmic showers
- Multi-track events contain a number of fake or badly reconstructed tracks
- Mostly low momentum tracks with a small number of hits and large ^{2}
- Fake rate is estimated to be around 1%.

Track reconstruction in Cosmic Runs

- Track reconstruction parameters for 1-track events
- Small discrepancies:
  - Detector noise
  - Simulation in low momentum range of cosmic ray muons (e.g. position of the concrete plug covering the shaft)

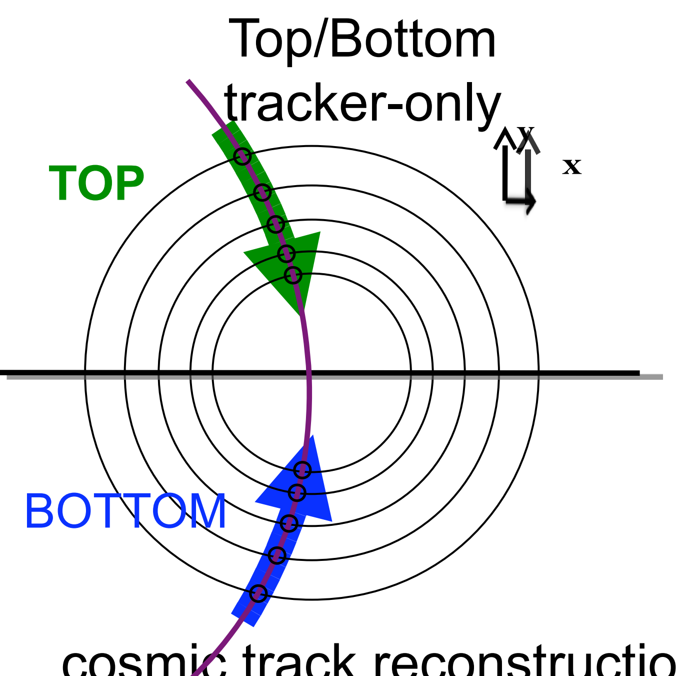




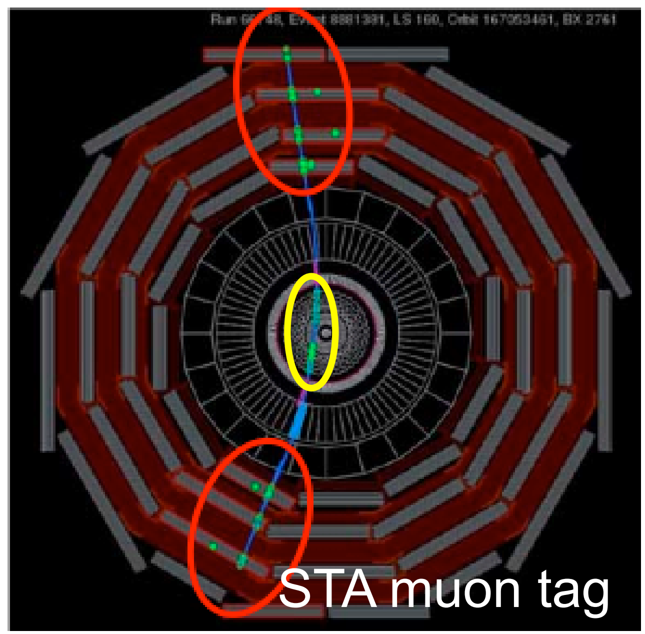


Track reconstruction efficiency in Cosmic Runs

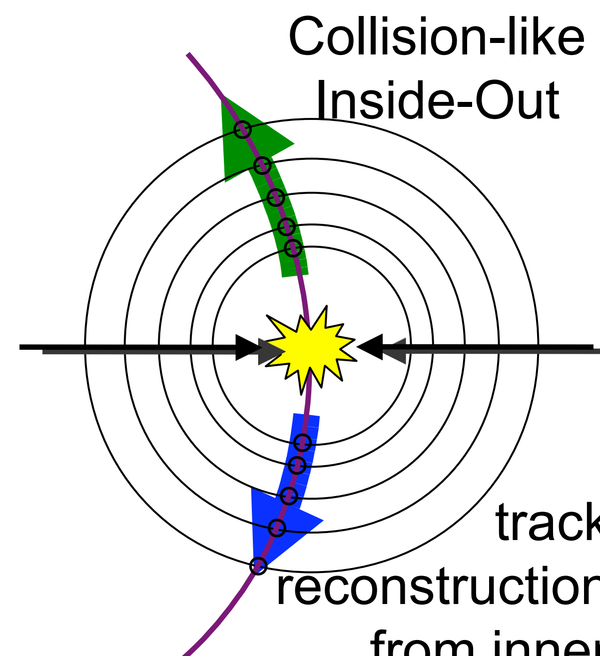
- 3 methods to estimate track reconstruction efficiency (barrel only) :
  1) Tag (standalone muon) and probe (tracker muon), collision like method
  2) Tracker-only, independent reconstruction of top and bottom muon legs
  3) Standard inside out seeding for collisions and two legs matching



cosmic track reconstruction in half of the tracker



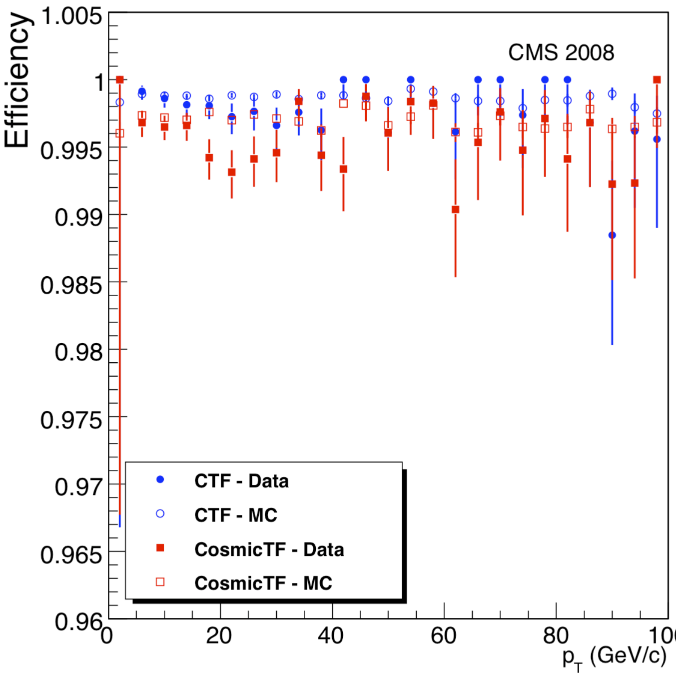
default cosmic track reconstruction

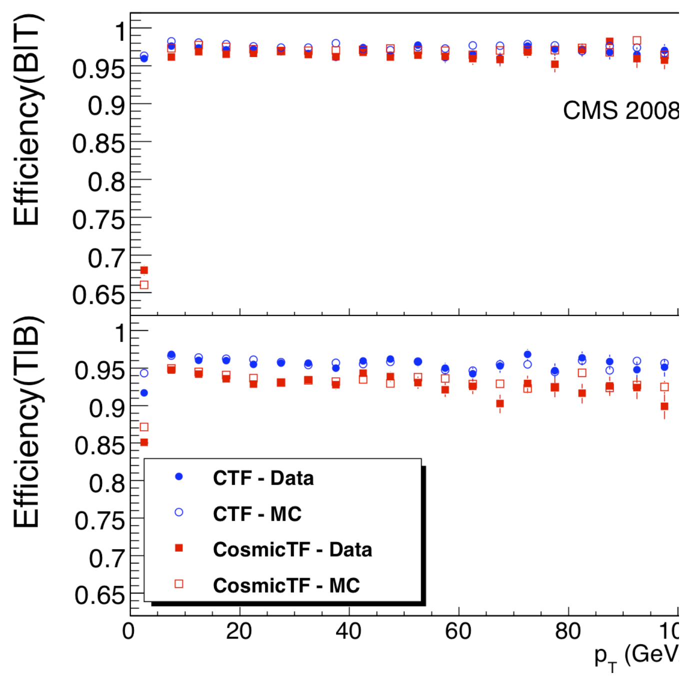


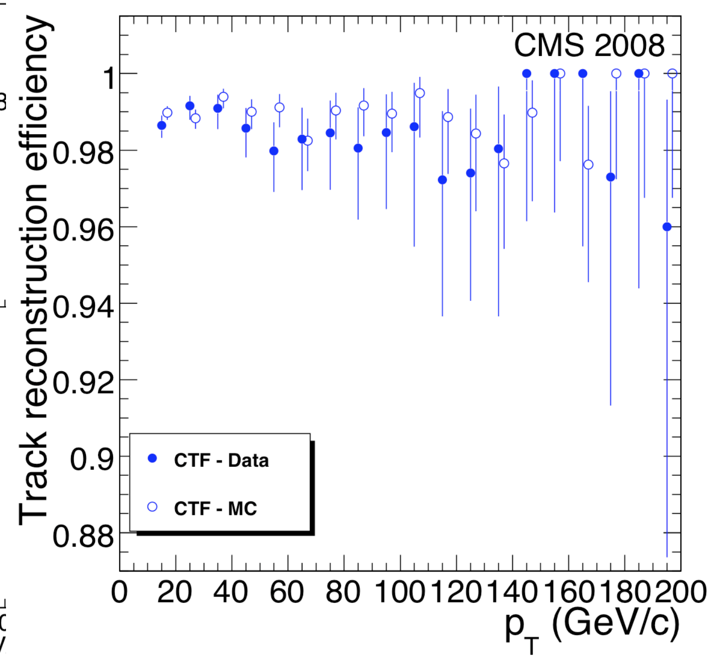
track reconstruction from inner seeds

Track reconstruction efficiency in Cosmic Runs

- 3 methods to estimate track reconstruction efficiency (barrel only) :
  1) Tag (standalone muon) and probe (tracker muon), collision like method
  2) Tracker-only, independent reconstruction of top and bottom muon legs
  3) Standard inside out seeding for collisions and two legs matching



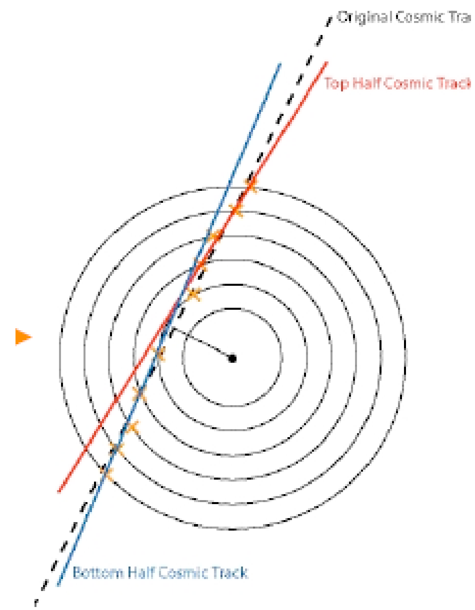


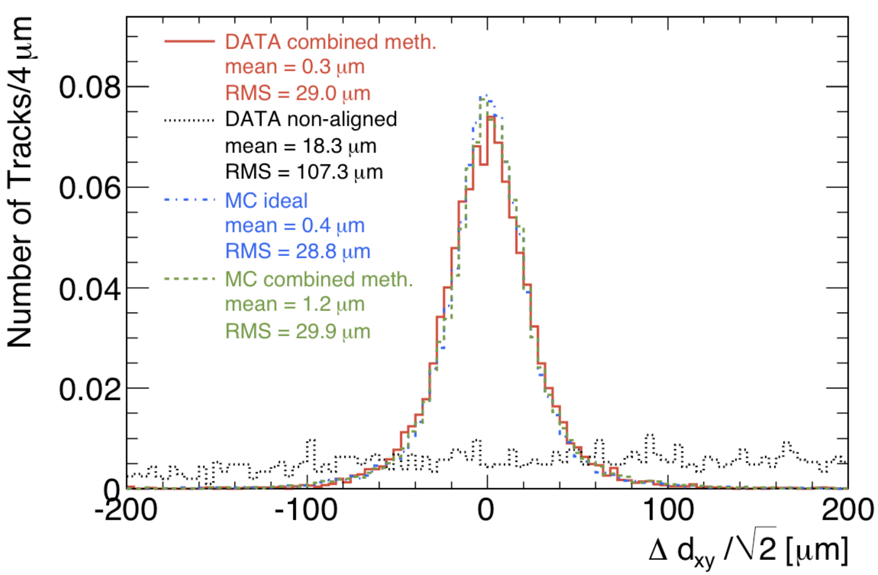


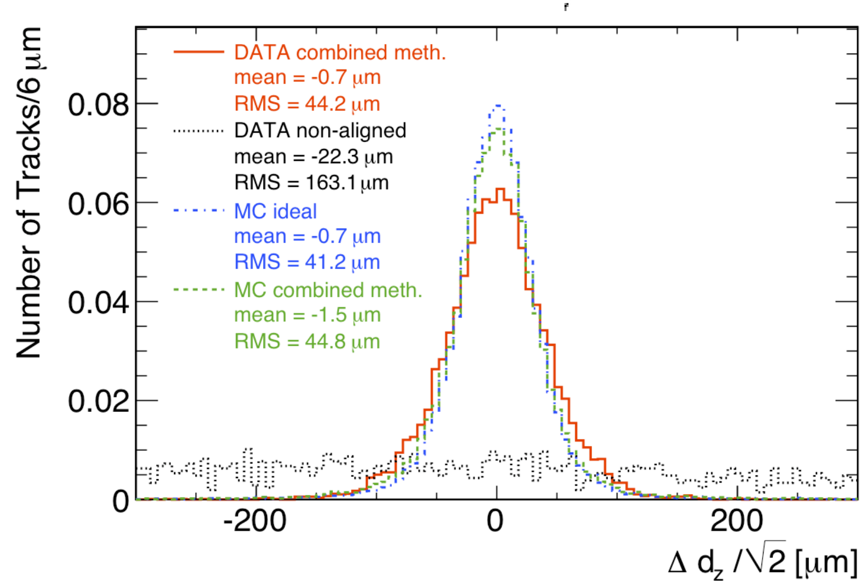
Collision-like reco:	Data	MC
Seeding	99.17 ± 0.12	99.30 ± 0.08
Pattern recognition	99.79 ± 0.06	99.64 ± 0.05
Overall	98.96 ± 0.13	98.94 ± 0.09

Track parameter resolution

- Splitting cosmic tracks along distance of closest approach to beamline
- Refit top and bottom halves and find the difference in the track parameters
- Validation of the alignment
  - Parameter resolution approaching those of a MC simulation with ideal detector geometry

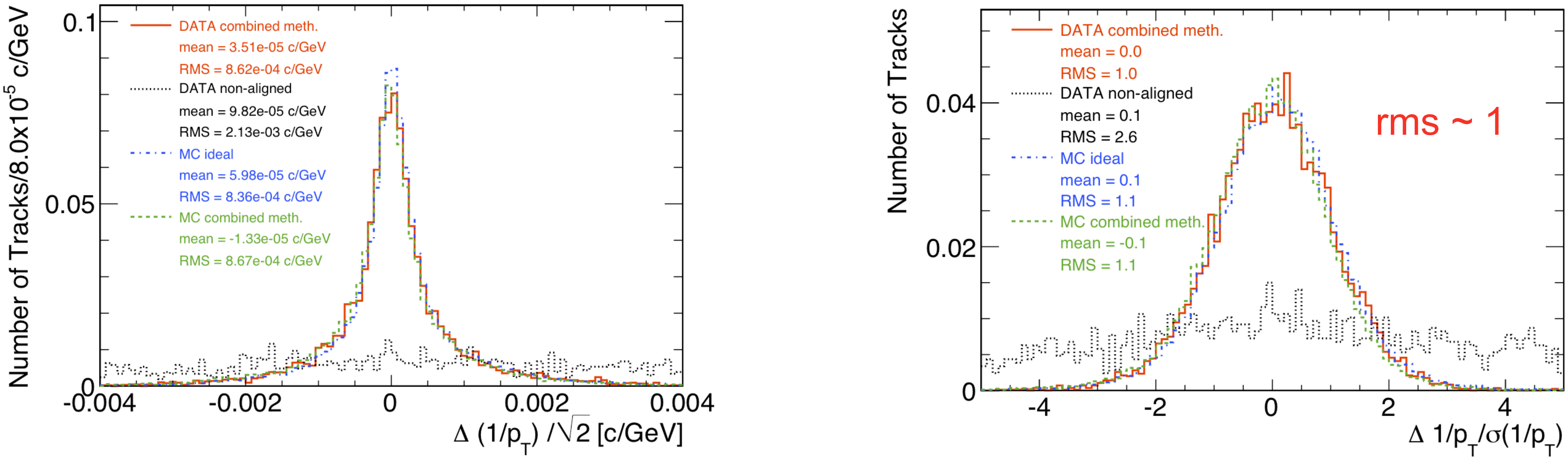




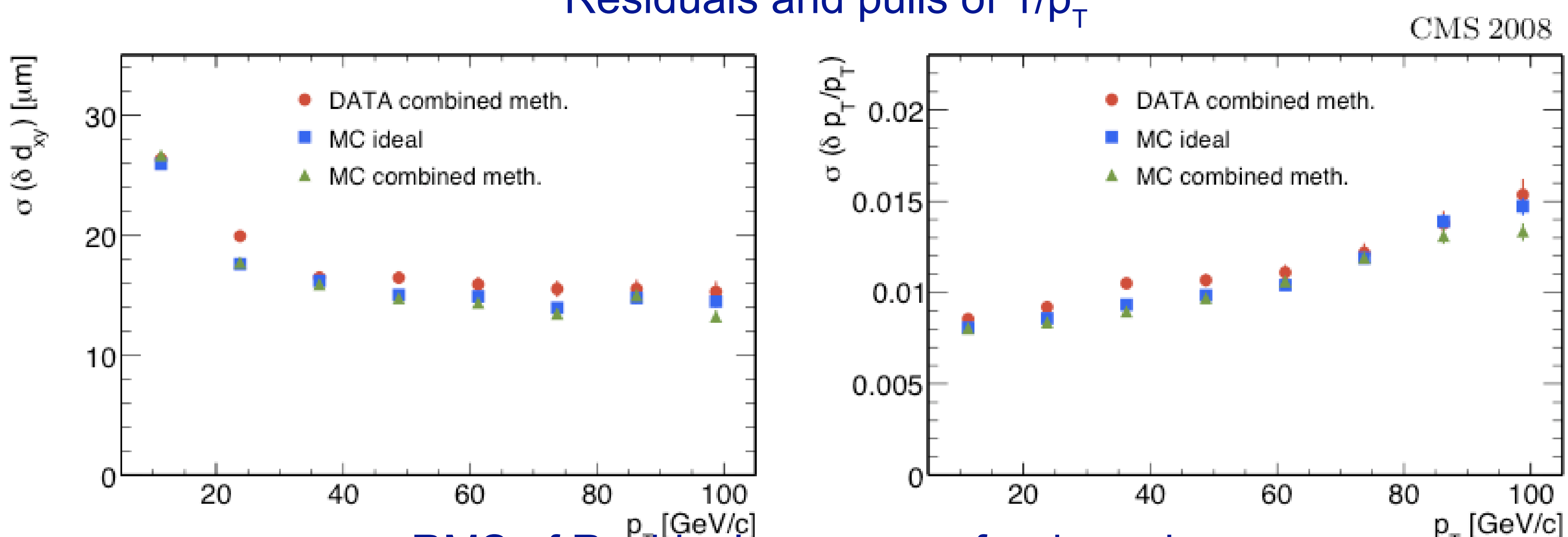


Residuals of transverse and longitudinal impact parameter

Track parameter resolution



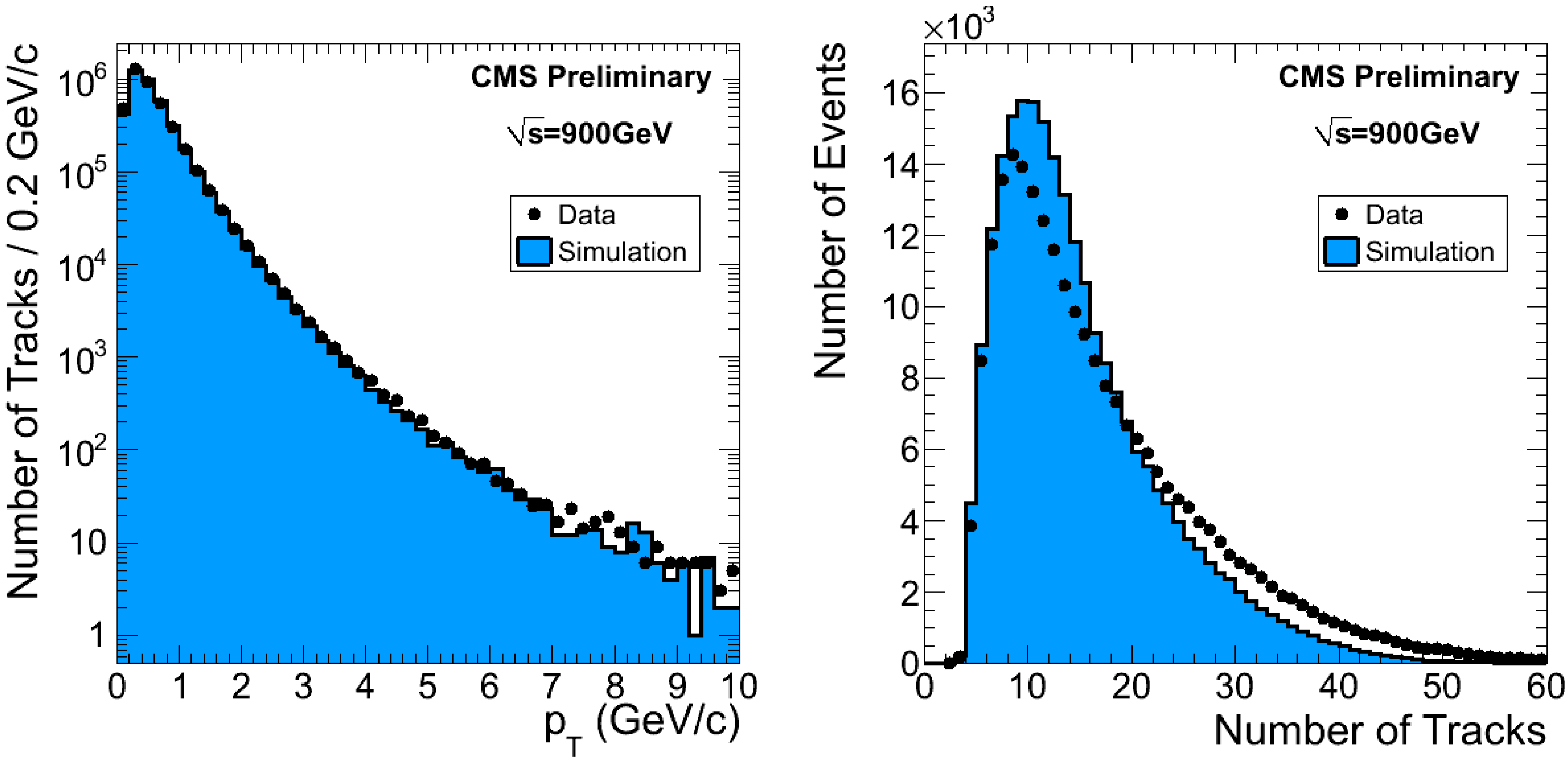
Residuals and pulls of $1/p_T$

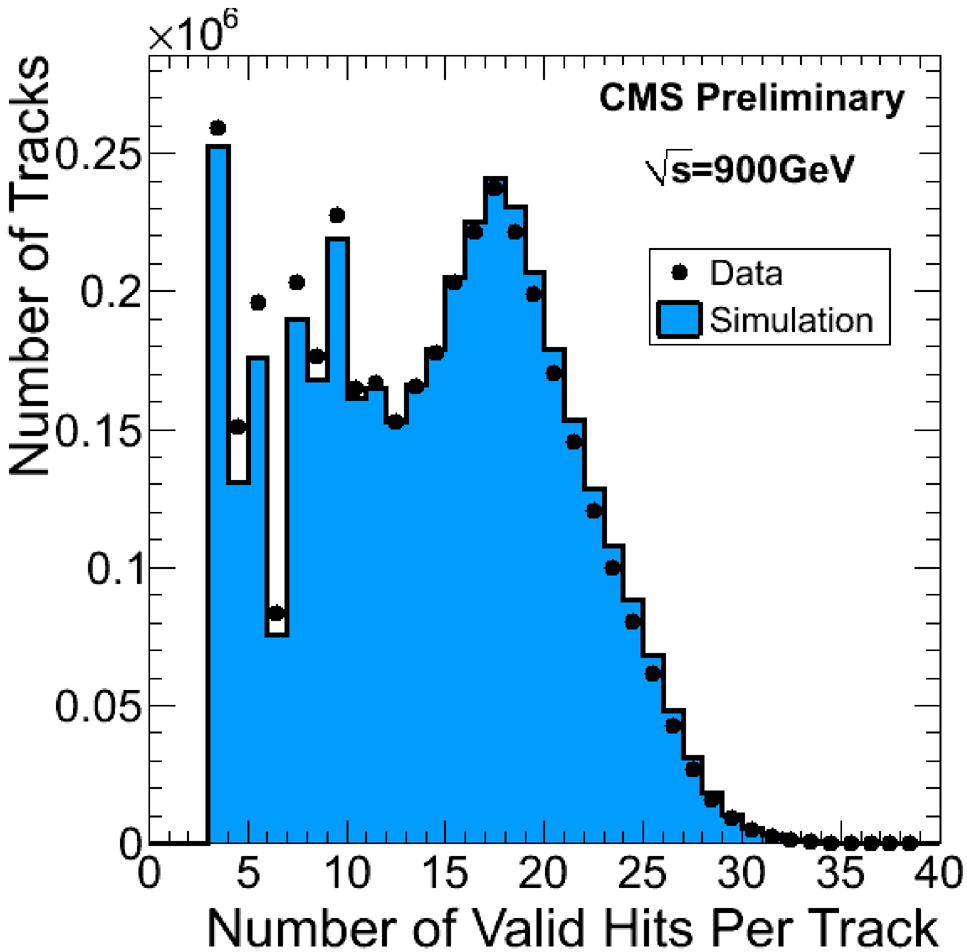


RMS of Residuals versus p_T for d_{xy} and p_T

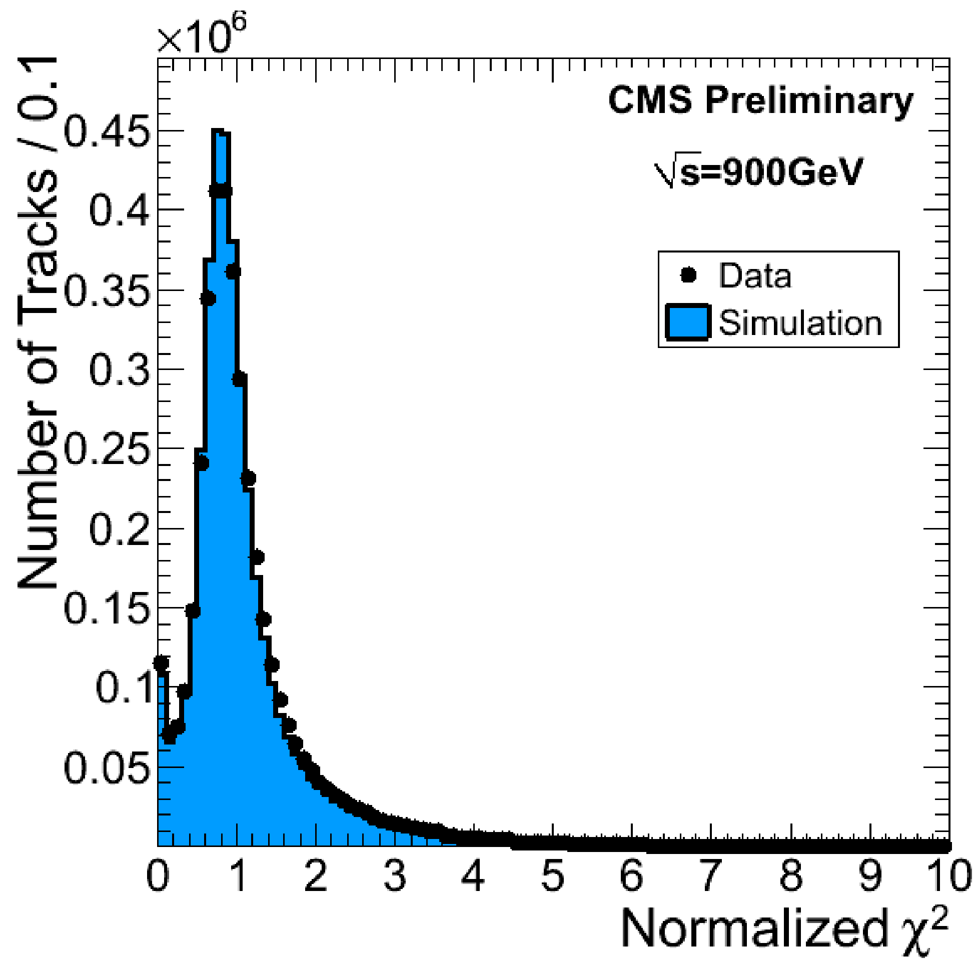
Track reconstruction in 900 GeV collisions

- Track reconstruction in pp collisions at 900 GeV (December 2009):



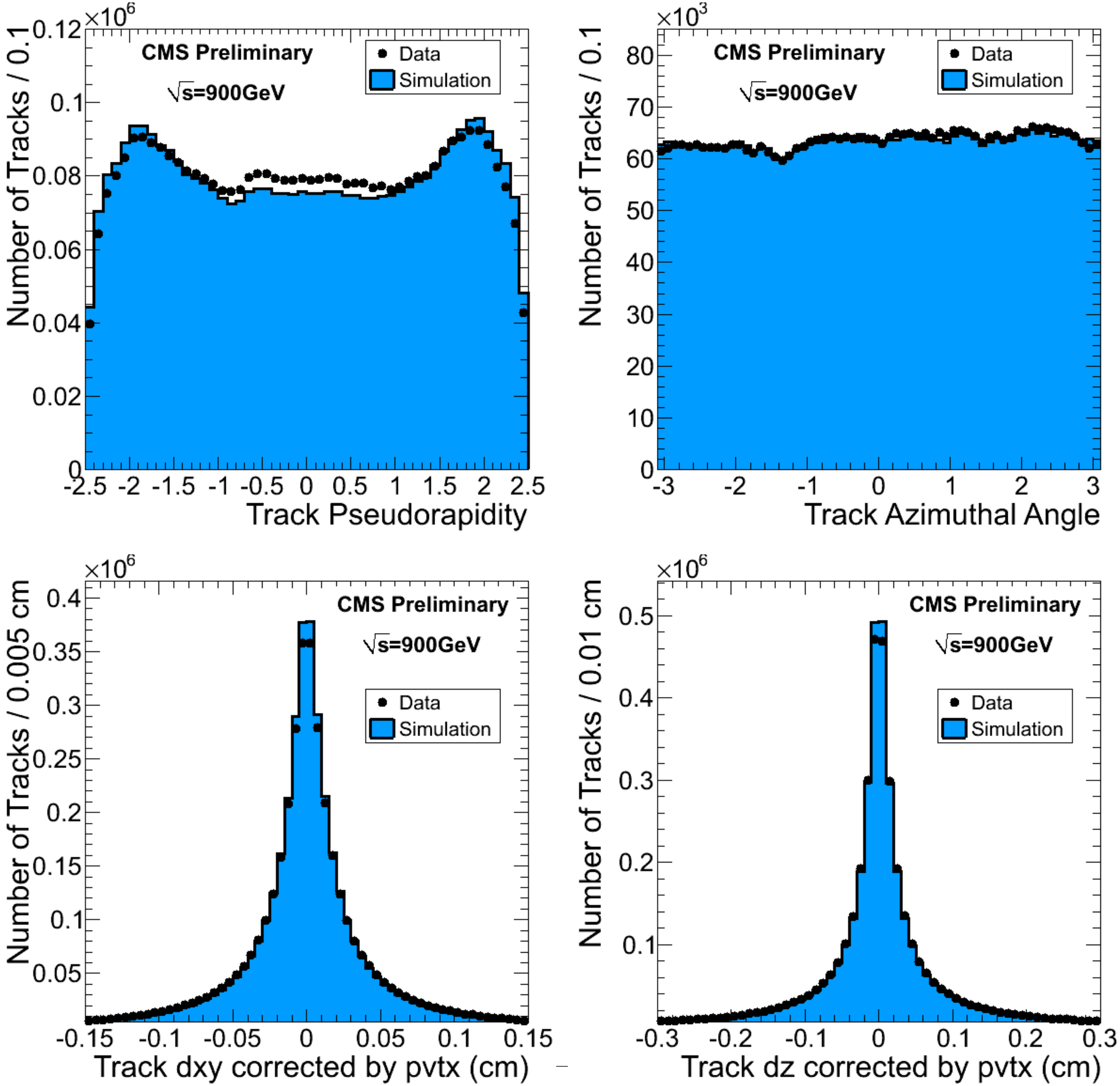


×10^6
CMS Preliminary
√s=900GeV
Data
Simulation
Number of Tracks
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40
Number of Valid Hits Per Track

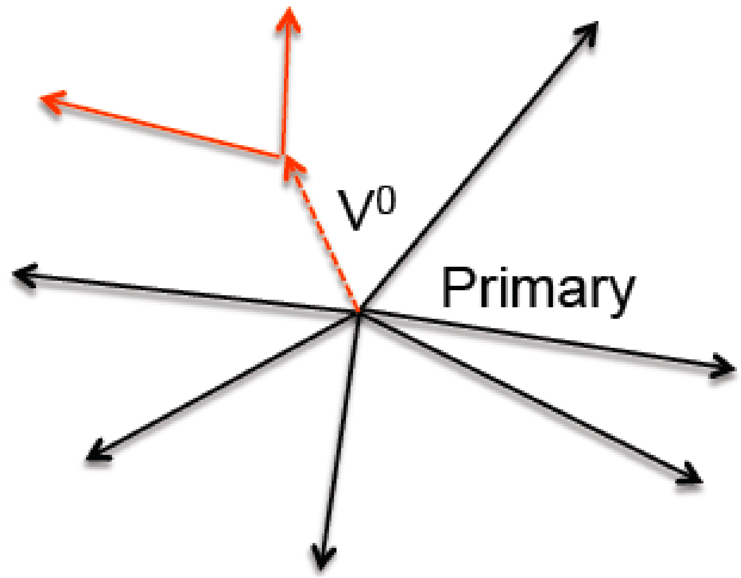


×10^6
CMS Preliminary
√s=900GeV
Data
Simulation
Number of Tracks / 0.1
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
5
6
7
8
9
10
Normalized χ²

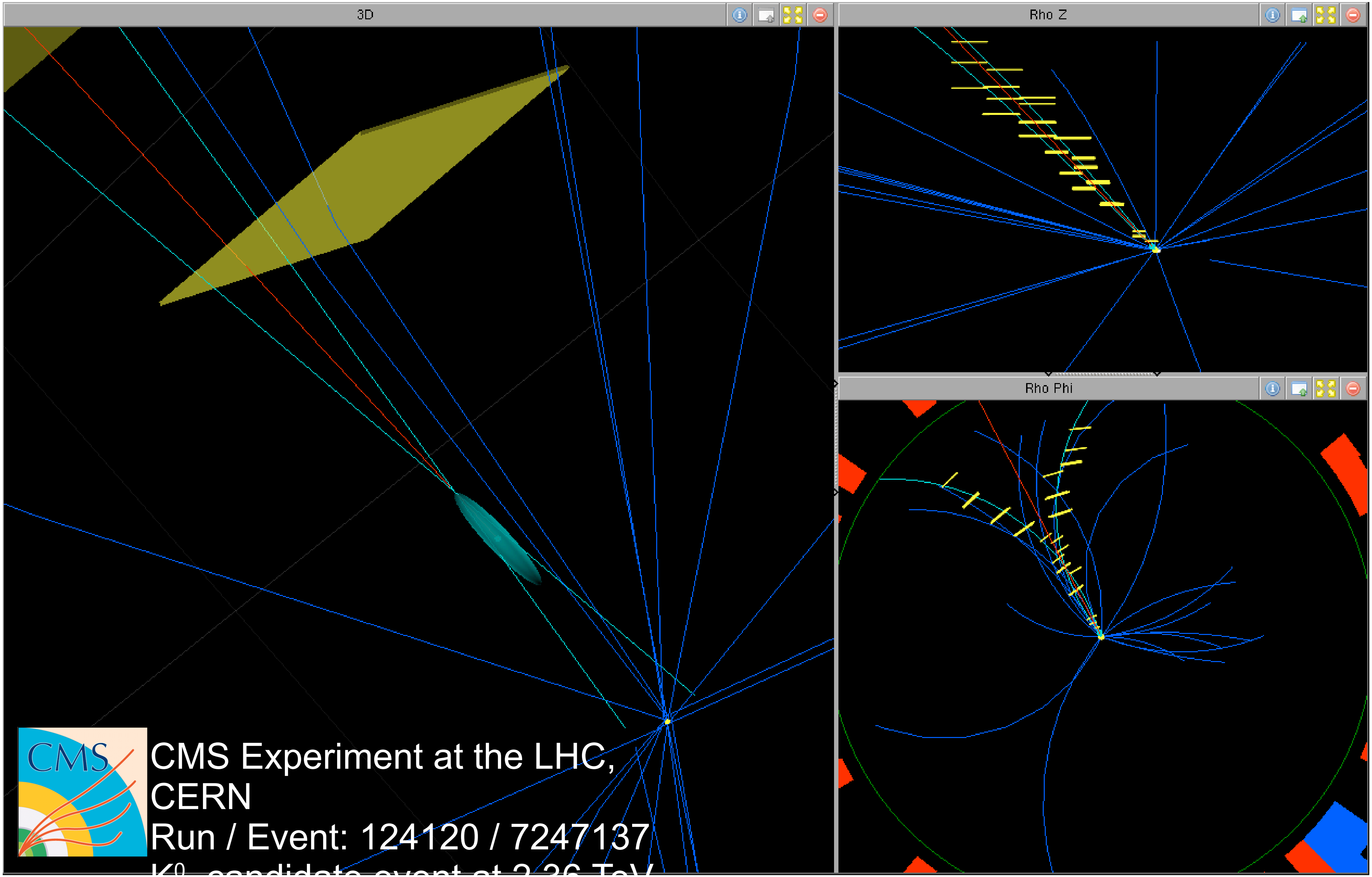
Track reconstruction in 900 GeV collisions



- Reconstruction of long-lived K^0_s and Λ :
  - $K_s^0 \to \pi^+\pi^-$
  - $\Lambda \to p\pi^-$
- Selection requirements
  - High quality oppositely charged tracks
  - Tracks not compatible with primary vertex
  - Displaced decay vertex
  - No track hits before the secondary vertex

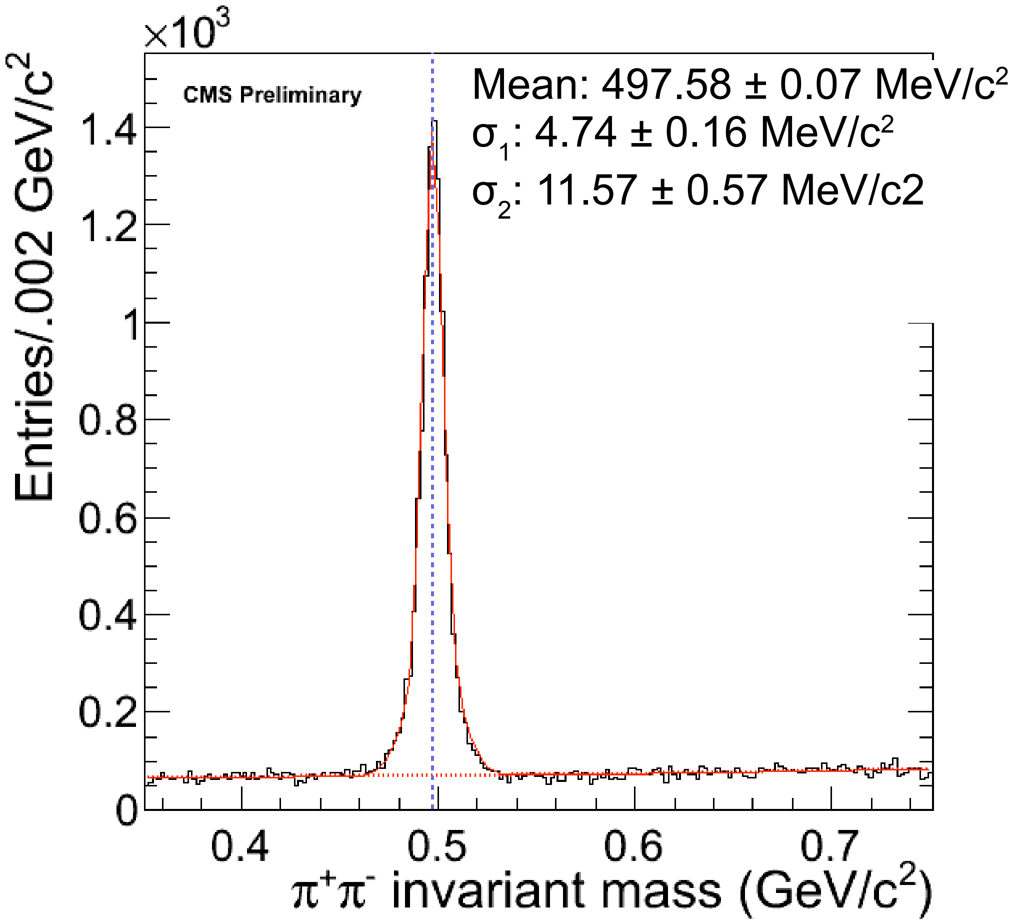


K^0_s candidate event at 2.36 TeV

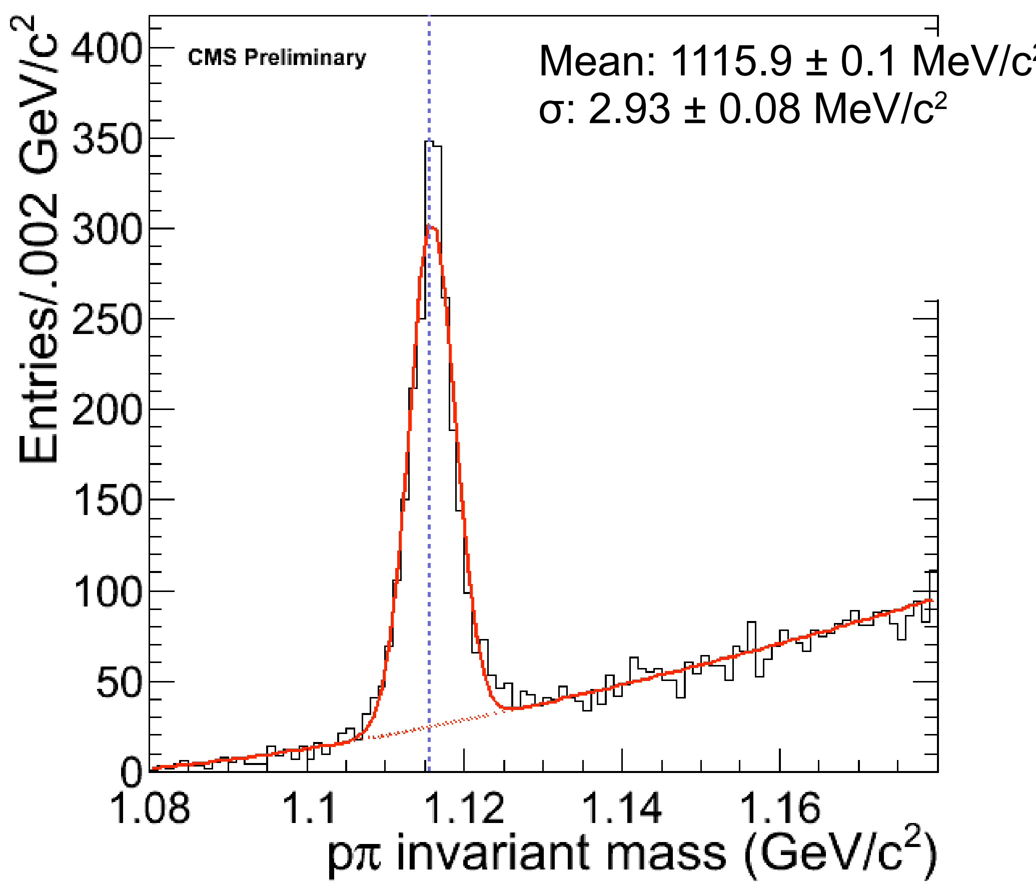


Reconstruction of long-lived K^0_s and Λ

Invariant mass of K^0_s and Λ candidates



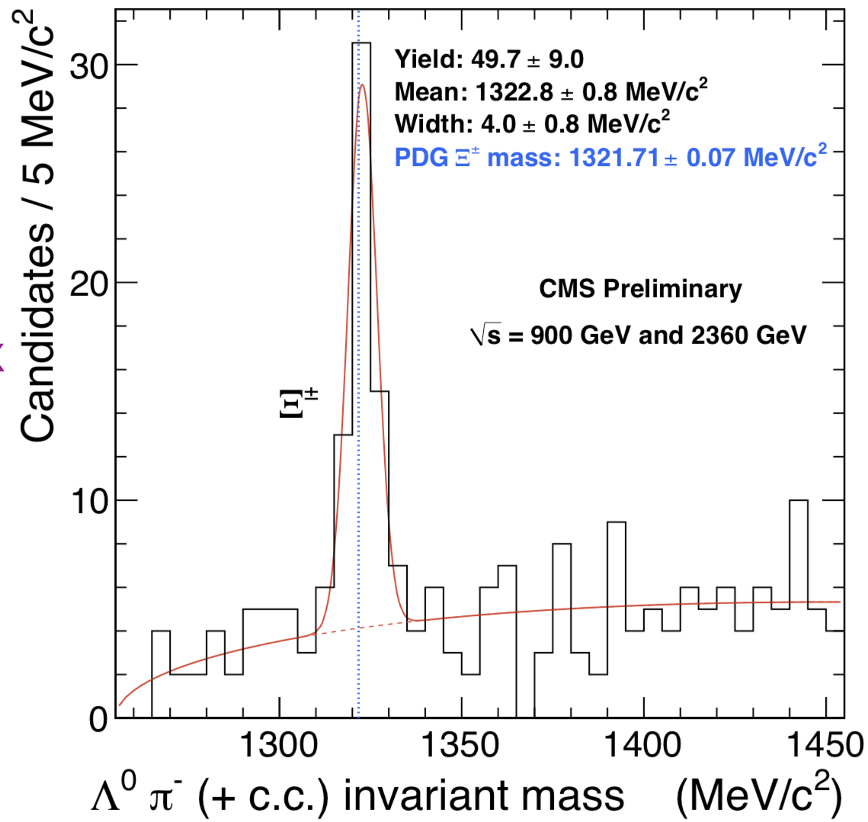
K^0_s signal peak fit with double Gaussian in good agreement with PDG mass (497.61 MeV/c²)

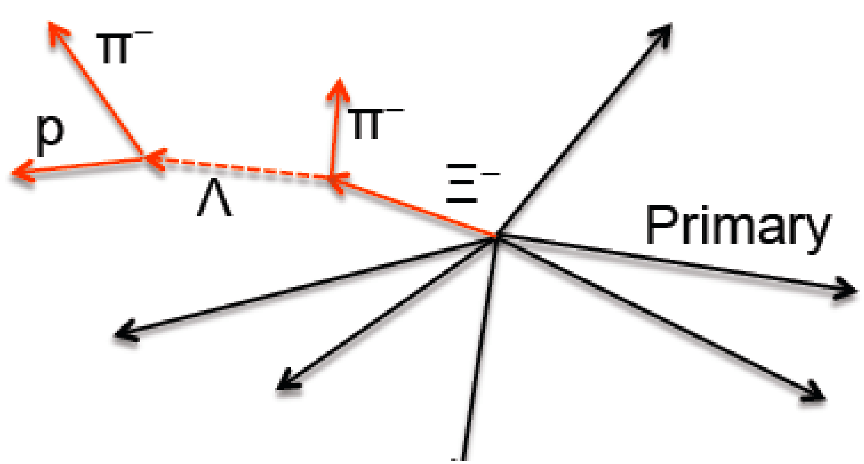


signal peak fit with single Gaussian in good agreement with PDG mass (1115.7 MeV/c²)

Reconstruction of Ξ⁻ baryon

- Reconstruction of Ξ⁻ baryon
  - Reconstruction of Λ
  - Vertex fit of Λ candidate with charged track,
  - Mass of Λ candidate constrained to PDG mass
  - Tracks not compatible with primary vertex
  - No track hits before the secondary vertex
- Clear mass peak consistent with Ξ⁻ production

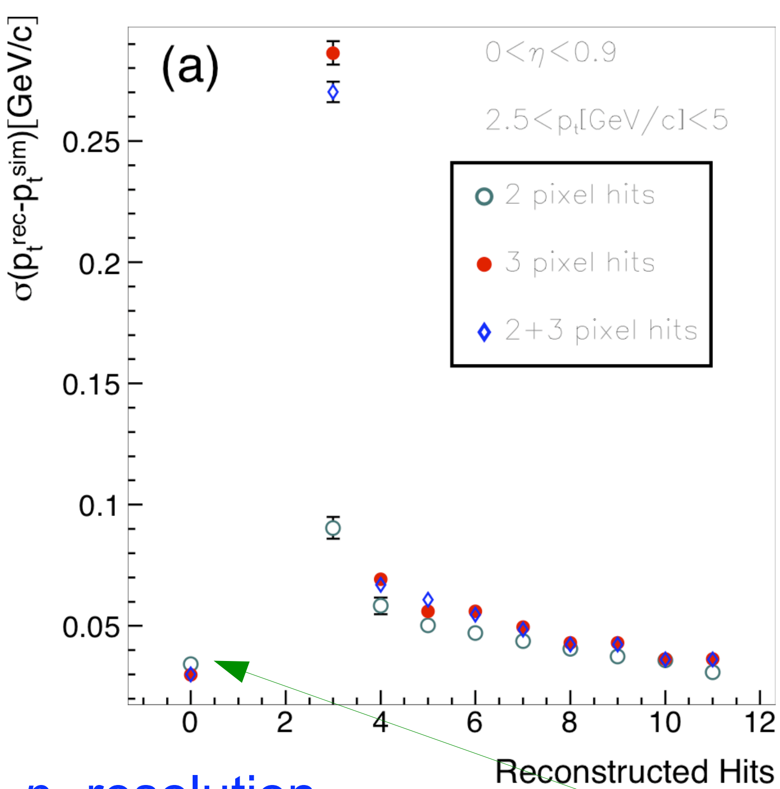




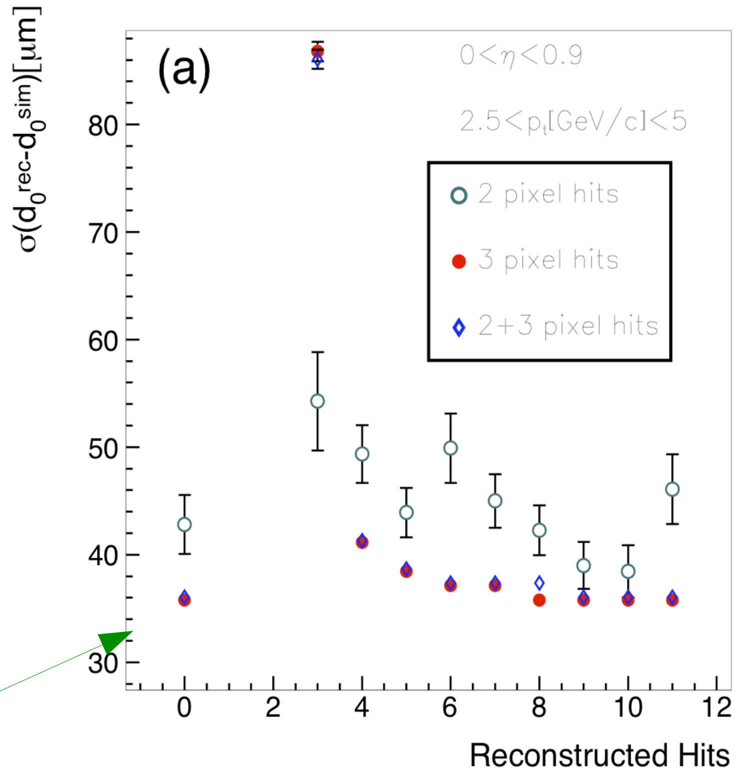
Partial reconstruction

- Combinatorial KF also suitable for usage in the High-Level Trigger (HLT):
  - Track parameter resolutions reach asymptotic value after using 5/6 hits

Resolutions as a function of the number of hits used: (b-jets, $2.5<p_T<5$, $|\eta|<0.9$)



p_T resolution

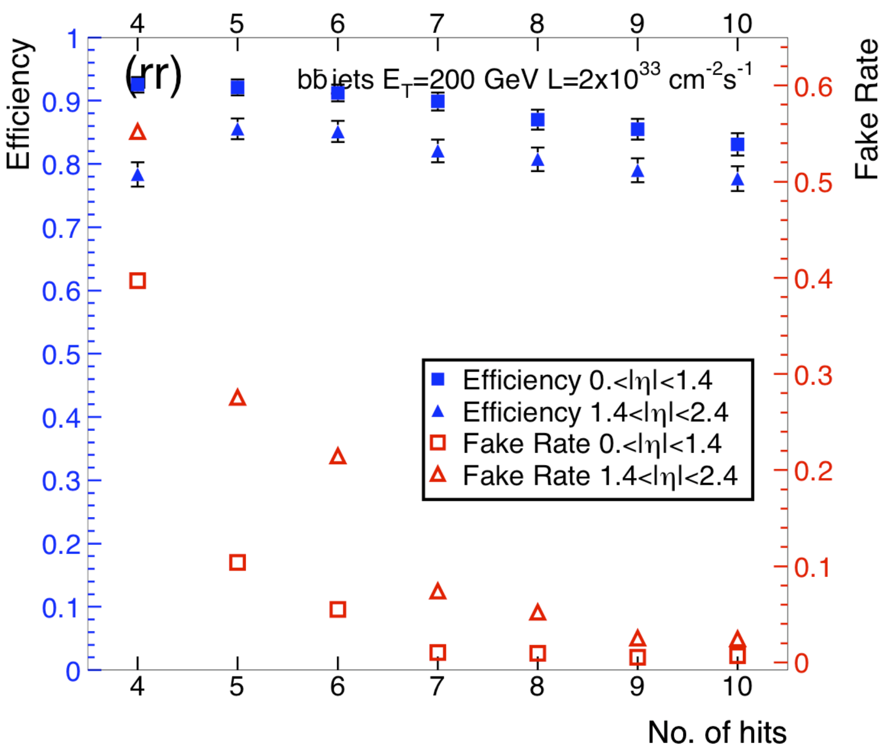


Transverse impact parameter resolution

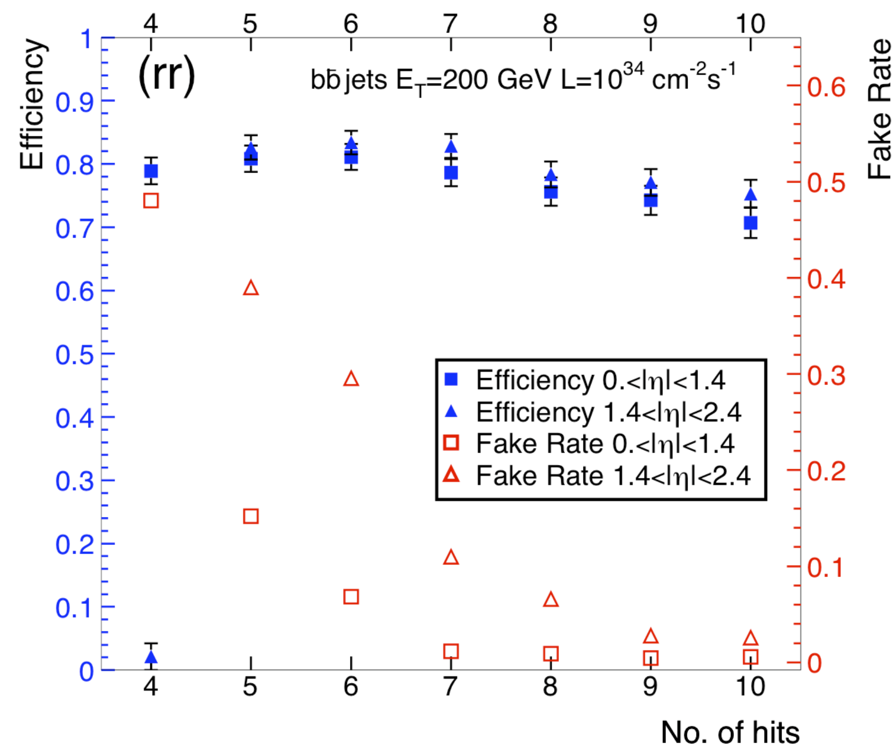
("0 hits" indicates full track reconstruction!)

Partial reconstruction

- Partial reconstruction: stop track reconstruction once enough information is available to answer a specific question
- Same components, algorithms used.
- Precision sufficient for most HLT applications (b-tagging)



b-jets, E_T = 200GeV, low lumi.



b-jets, E_T = 200GeV, high lumi.

Adaptive methods

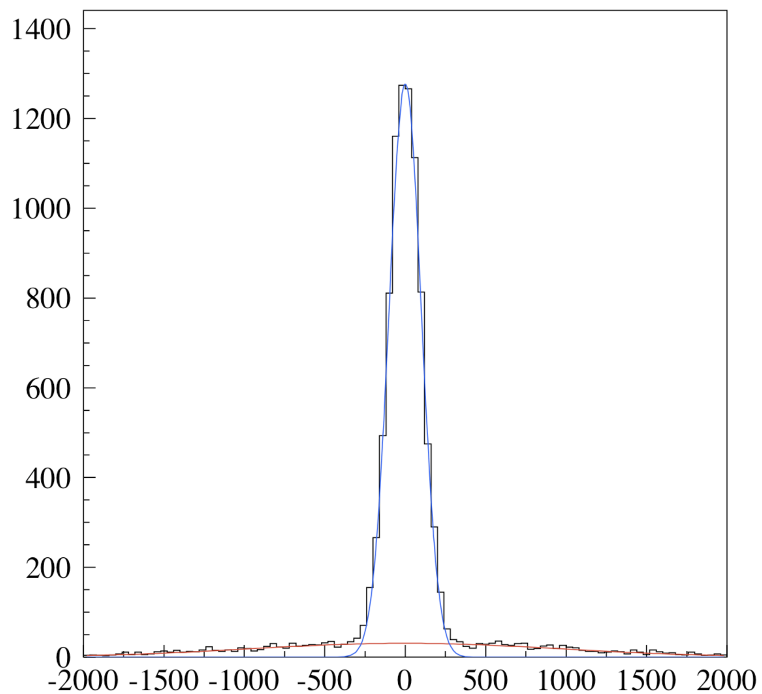
- Several non-linear algorithms have been implemented
- LSM optimal when
  - model is linear
  - random noise Gaussian
- Pdf involved are usually non-Gaussian:
  - Measurement errors have Gaussian core, with tails
  - Energy loss and multiple scattering (tails)

  → Gaussian-sum Filter
- Large background noise (electronic noise, low p_T tracks, δ electrons...)
  - Hit degradation
  - Hit assignment errors

  → Deterministic Annealing Filter & Multi-Track Fit

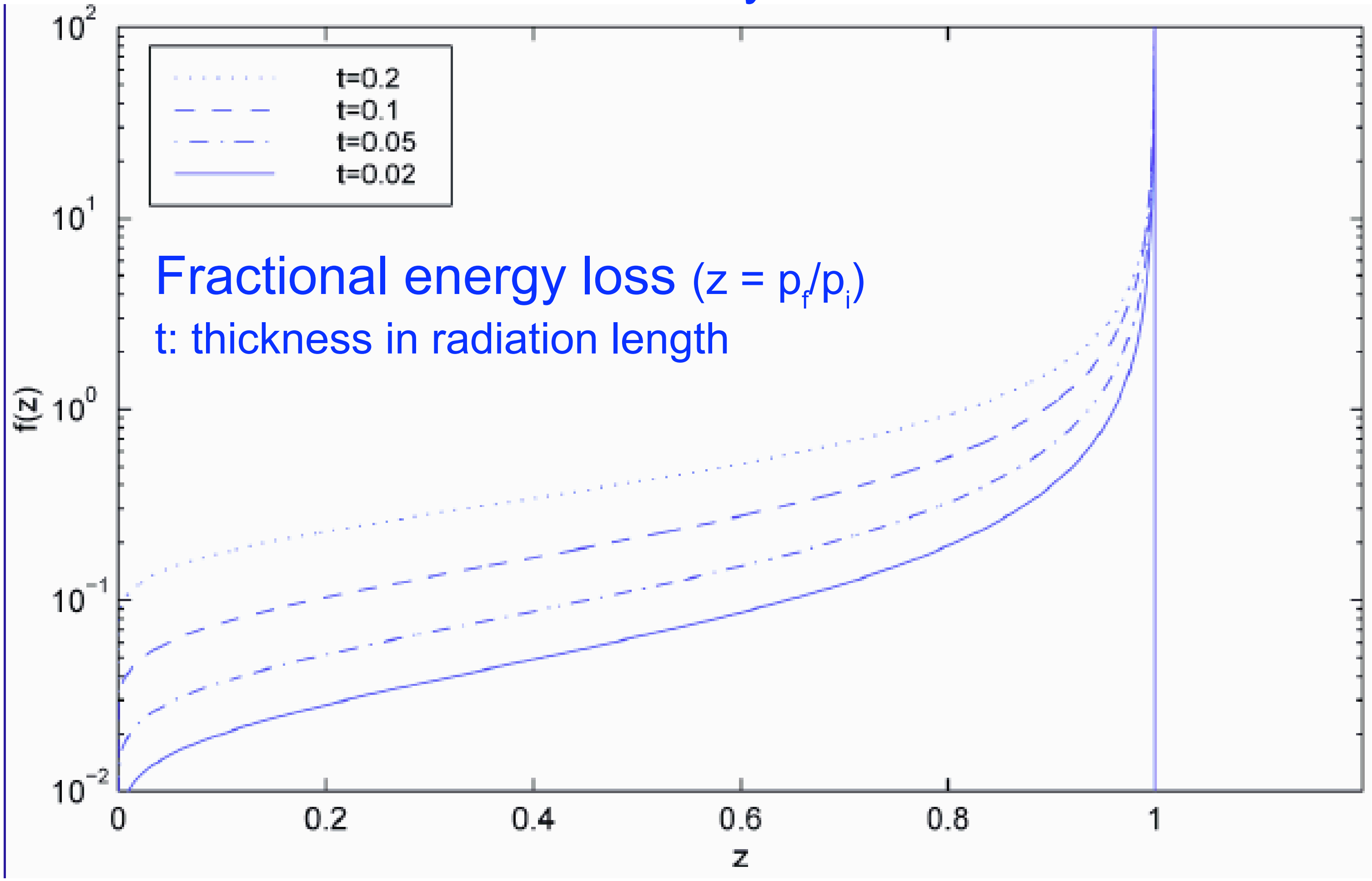
The Gaussian-sum Filter

- Pdf involved are usually non-Gaussian:
  - Measurement errors have Gaussian core, with tails
  - Energy loss and multiple scattering (tails)
- Gaussian-sum Filter (GSF): instead of single Gaussian, model the pdfs involved by mixture of Gaussians:
  - Main component of the mixture would describe the core of the distribution
  - Tails would be described by one or several additional Gaussians.



The Gaussian-sum Filter

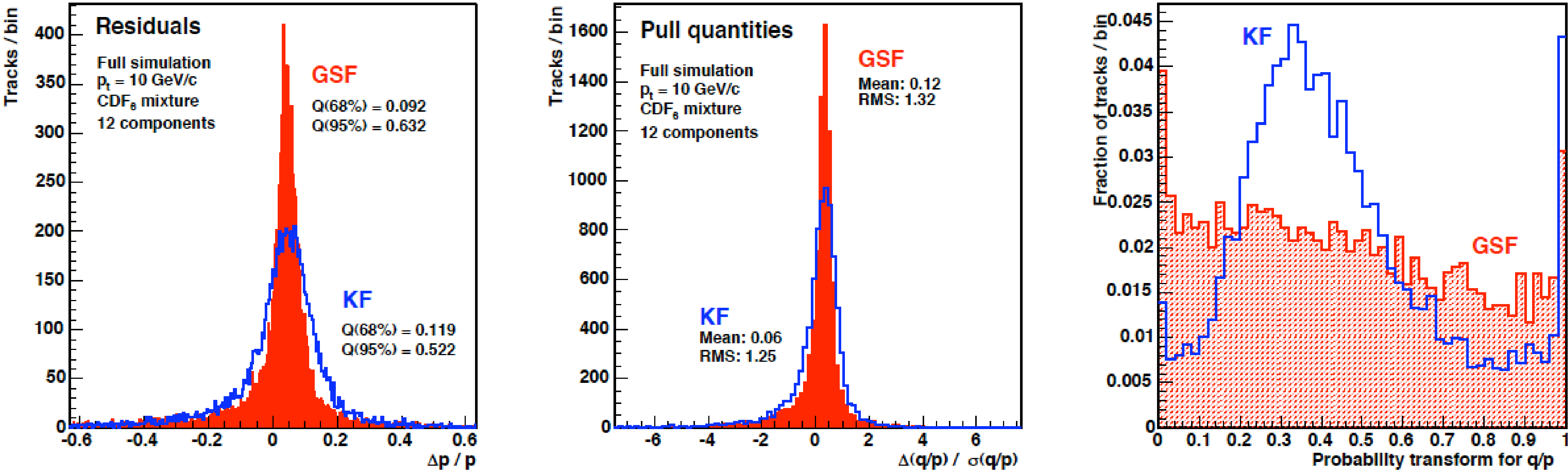
- For electrons, above ~100MeV/c, energy loss dominated by bremsstrahlung
  - Bethe and Heitler energy loss model is highly non-Gaussian
  - In the standard KF, distribution approximated by single Gaussian
- Model the Bethe-Heitler distribution by a mixture of Gaussians



The Gaussian-sum Filter

- All involved distributions are Gaussian mixtures
- State vector is also distributed according to a mixture of Gaussians
- GSF: Non-linear generalization of the Kalman Filter
- Weighted sum of several Kalman Filters
  - GSF is implemented as a number of Kalman filters run in parallel
  - The weights of the components are calculated separately
- Estimator is non-linear: weights depend on the measurements
- A pseudo-χ^2 can be defined, but it is not χ^2 distributed
- Exponential growth: combinatorial combination of the state vector components with energy-loss components
  - Number of components have to be limited to a predefined number at each step
  - Cluster (collapse) components with the smallest 'distance' (Distance measurements: Kullback-Leibler Distance or Mahalanobis Distance)
- Output is full Gaussian mixture of state vector
  - Can be used in subsequent application (GSF vertex fit)

The Gaussian-sum Filter

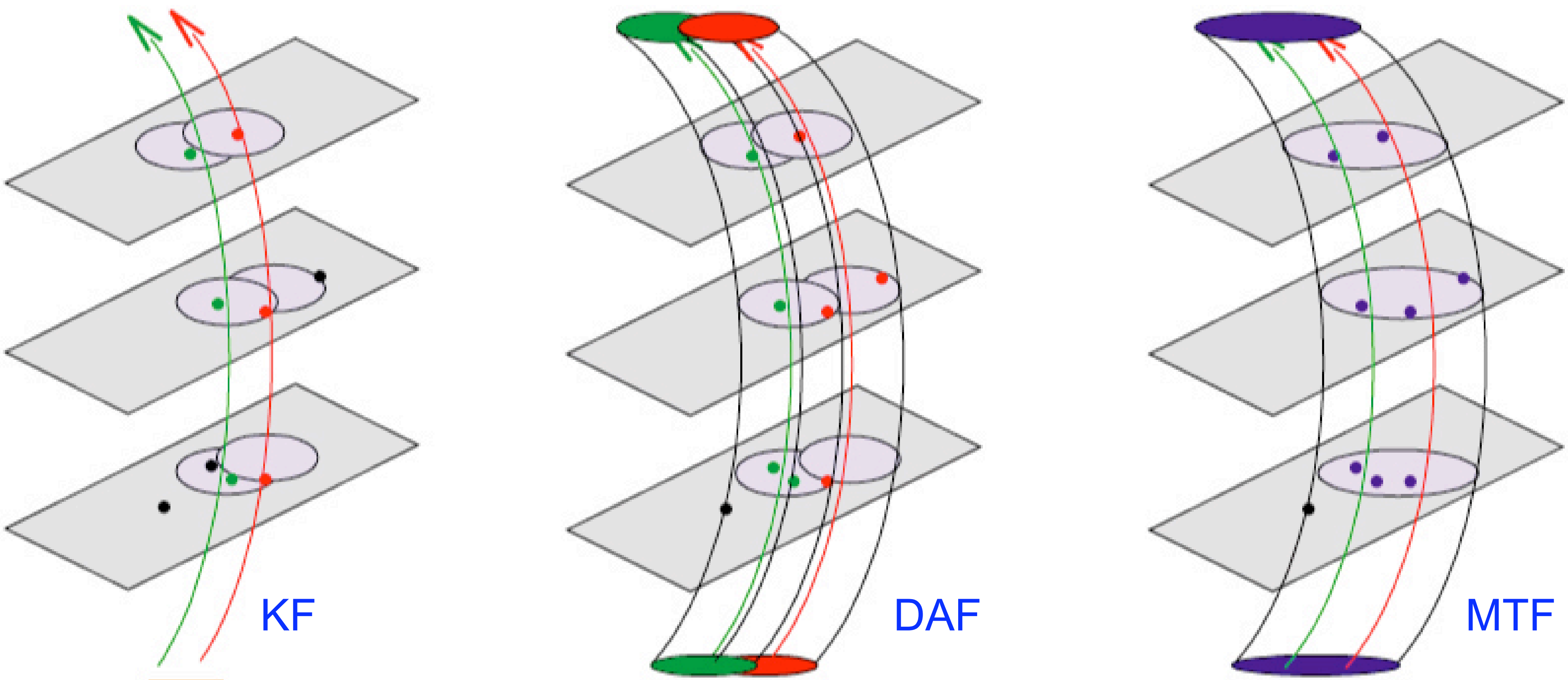


- Improvement of the core of the residual distribution
- Little reduction of the tails:
  - Radiation in the first layer can not be detected
    - can be compensated by vertex constraint
  - Non-Gaussian measurement errors in the Pixel detectors
    - Incorporate Gaussian mixtures of measurement errors (also for non-e fits)
- Most efficient for low energy electrons (a few tens of GeV), little gain at 100 GeV

Adaptive filters: the DAF and the MTF

- In very dense environments (e.g. high E_T *b*-jets, τ jets), degradation due to large background noise
  - High track density: hit degradation due to contamination of nearby tracks
  - High hit density: wrong hit assignment
- Kalman Filter: hard hit assignment
- Soft hit assignment may be more suitable
- Global approach of hit assignation, using full track information
  - Part of the hit assignment done in the final track fit

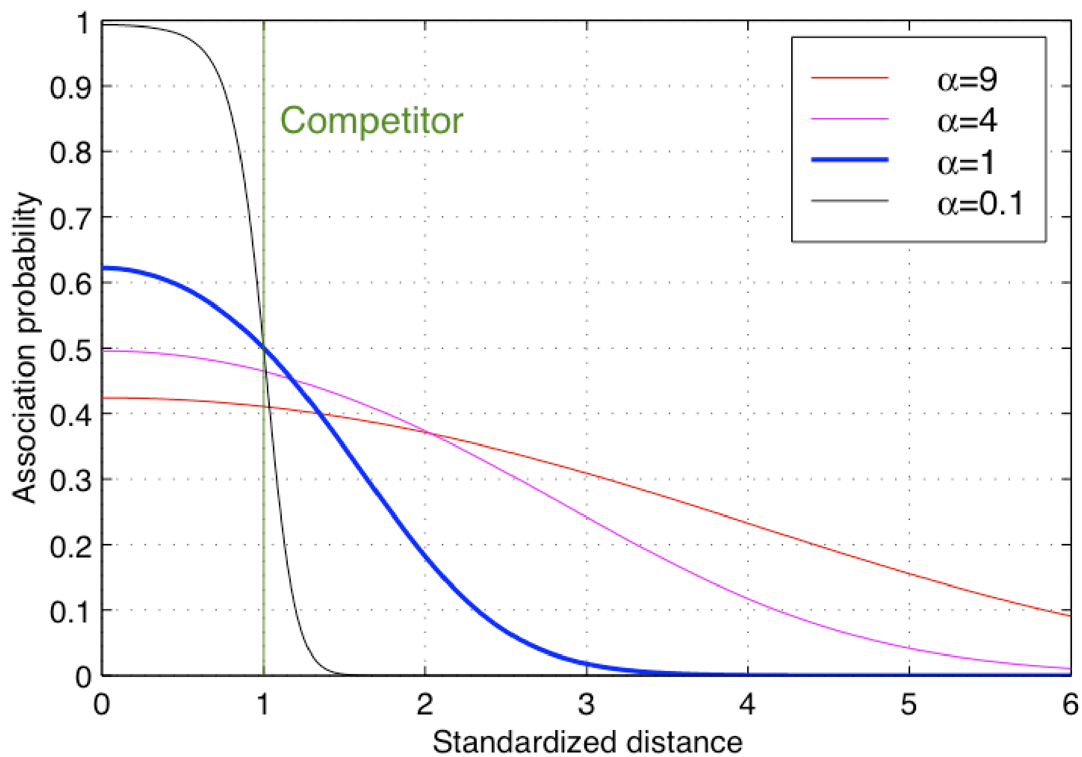
Adaptive filters: the DAF and the MTF



- Deterministic Annealing Filter (DAF): single track fit
  - On a same surface, several hits may compete for a track
- Multi-Track Fit (MTF): concurrent multi-track fit on collection of hits
  - Competition between tracks and hits
  - Each hit on a layer can belong to each of several tracks

Adaptive filters: the DAF and the MTF

- Deterministic Annealing Filter (DAF): single track fit
  - Competition between hits: on a same surface, several hits may compete for a track
  - Hit weights (assignment probability) based on hit-track distance (residual) and competing measurements



Hit weight in the presence of a competitor. The competitor is at one standard deviation from the track.

Adaptive filters: the DAF and the MTF

- Deterministic Annealing Filter (DAF): single track fit
  - Competition between hits: on a same surface, several hits may compete for a track
  - Hit weights (assignment probability) based on hit-track distance (residual) and competing measurements
- Multi-Track Fit (MTF): concurrent multi-track fit on collection of hits
  - Competition between tracks and hits
  - Each hit on a layer can belong to each of several tracks
  - Hit weights based on hit-track distance and competing measurements and tracks

Adaptive filters: the DAF and the MTF

- Deterministic Annealing Filter (DAF): single track fit
  - Competition between hits: on a same surface, several hits may compete for a track
  - Hit weights (assignment probability) based on hit-track distance (residual) and competing measurements
- Multi-Track Fit (MTF): concurrent multi-track fit on collection of hits
  - Competition between tracks and hits
  - Each hit on a layer can belong to each of several tracks
  - Hit weights based on hit-track distance and competing measurements and tracks
- Iterative Kalman Filter with annealing
  - Full Kalman fit (filter+smoother), using the current weights
  - Calculation of weights, using current estimates
  - The iteration ends when the weights are stable
- Deterministic Annealing helps to reach the optimal solution
  - Final assignment probabilities may depend on initial values
  - At the start $T >> 1$
  - At each iterations, T reduced according to a predefined schedule, until $T = 1$

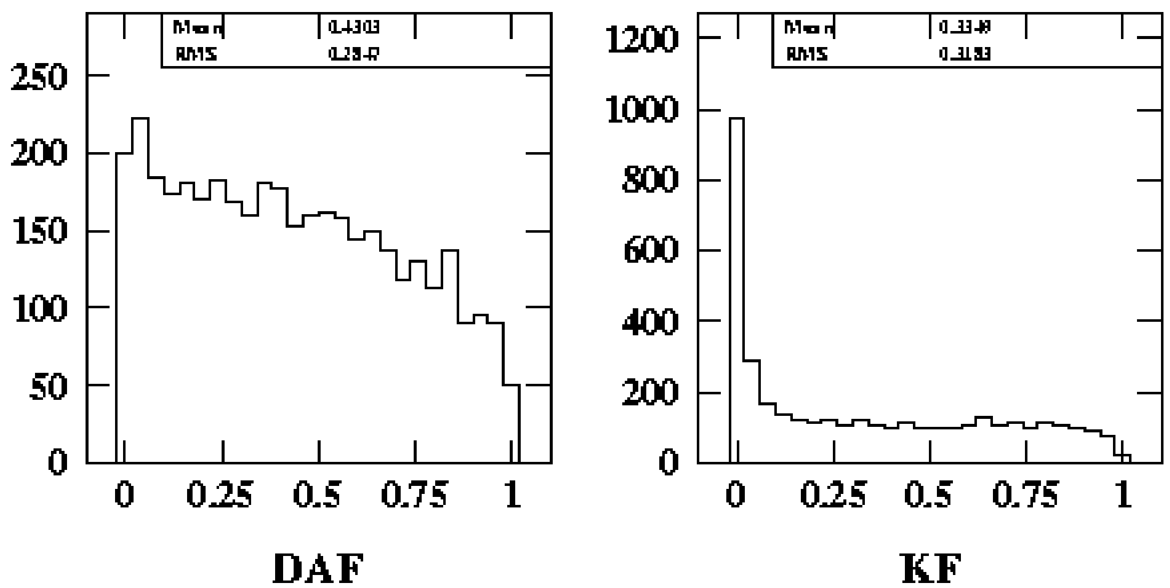
BROWN

Adaptive filters: the DAF and the MTF

- Both need initial hit collection and track seed(s): basic pattern recognition and track parameters from KF
  - DAF: initial hit collection around a KF track.
  - MTF: collection of tracks from KF (or even DAF), close in momentum space, hits collected around these tracks
- With this seeding, track finding efficiencies can not be improved w.r.t. KF

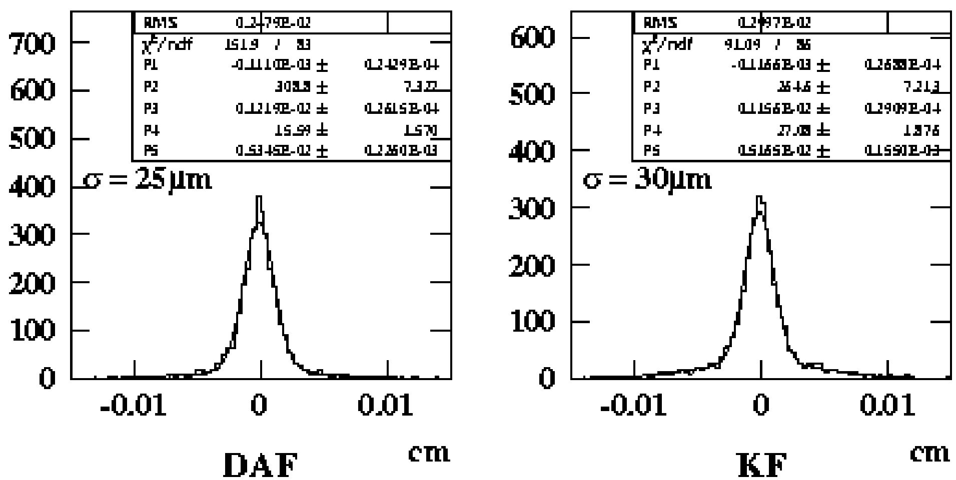
- For “isolated tracks”, even at high luminosity, the DAF does not provide a measurable improvement in track quality
- “dense environment”: b-jet with E_T=200 GeV,
  - Tracks with p_T > 15 GeV/*c*, min. 8 hits:
  - Better track quality (χ^2)

χ^2 probability - $|\eta|<0.7$

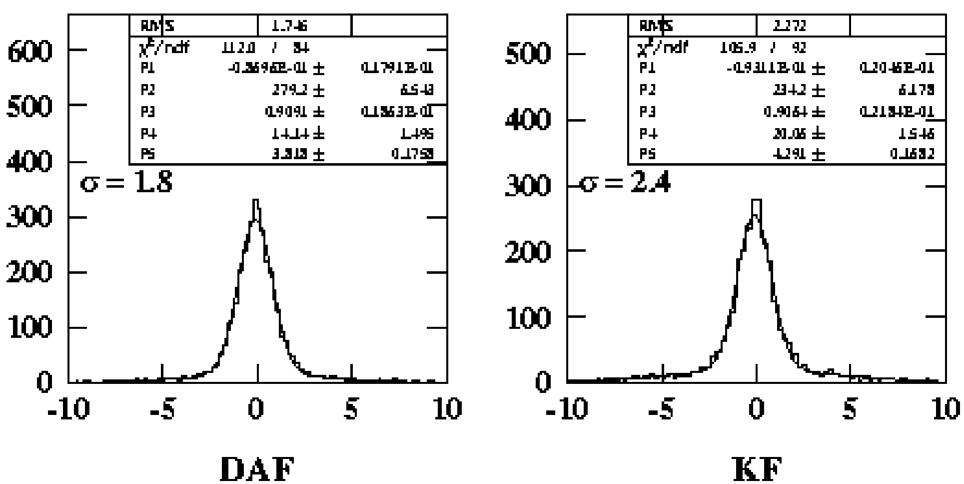


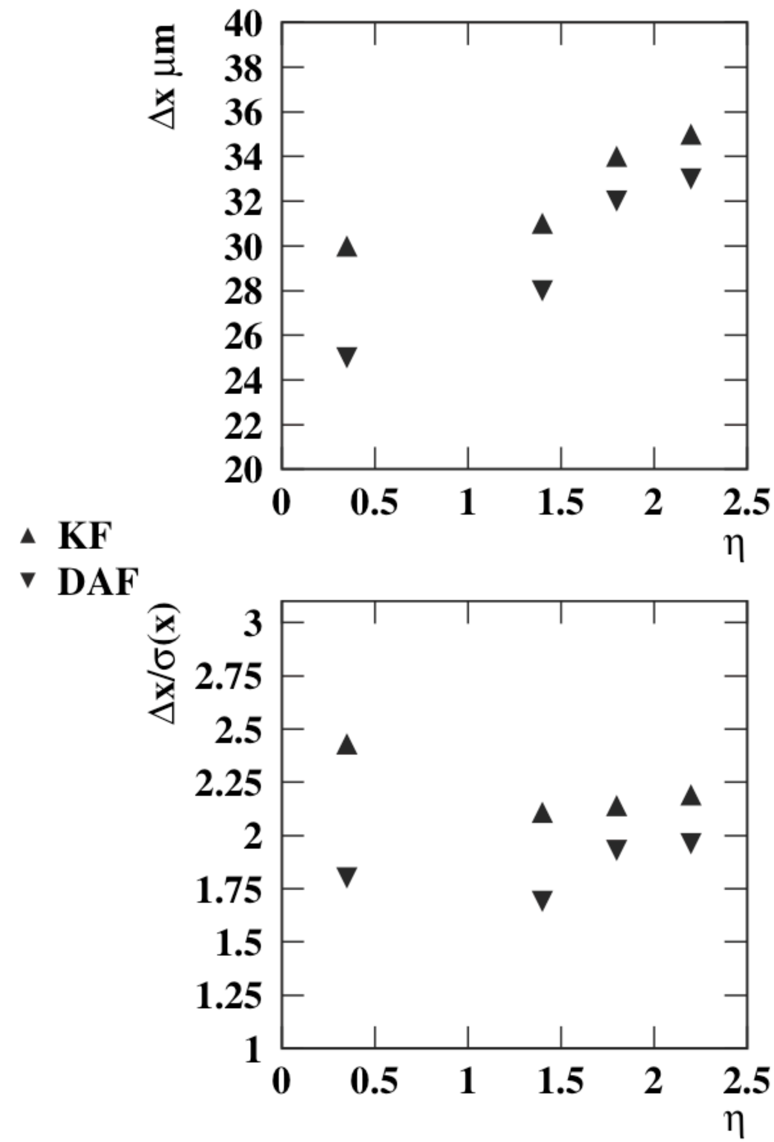
The Deterministic Annealing Filter

Transverse IP resolution - |η|<0.7



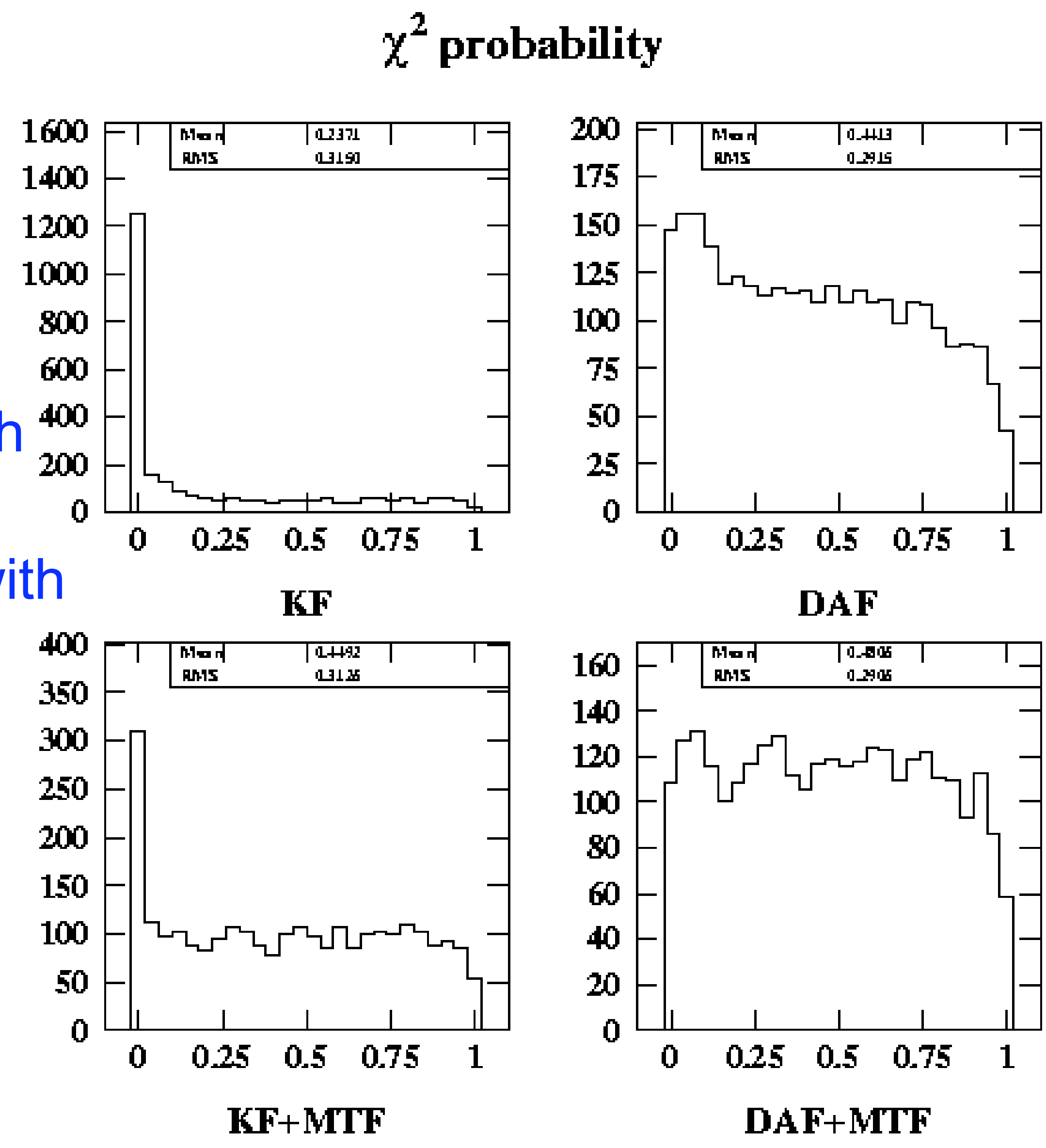
Transverse IP pull - |η|<0.7





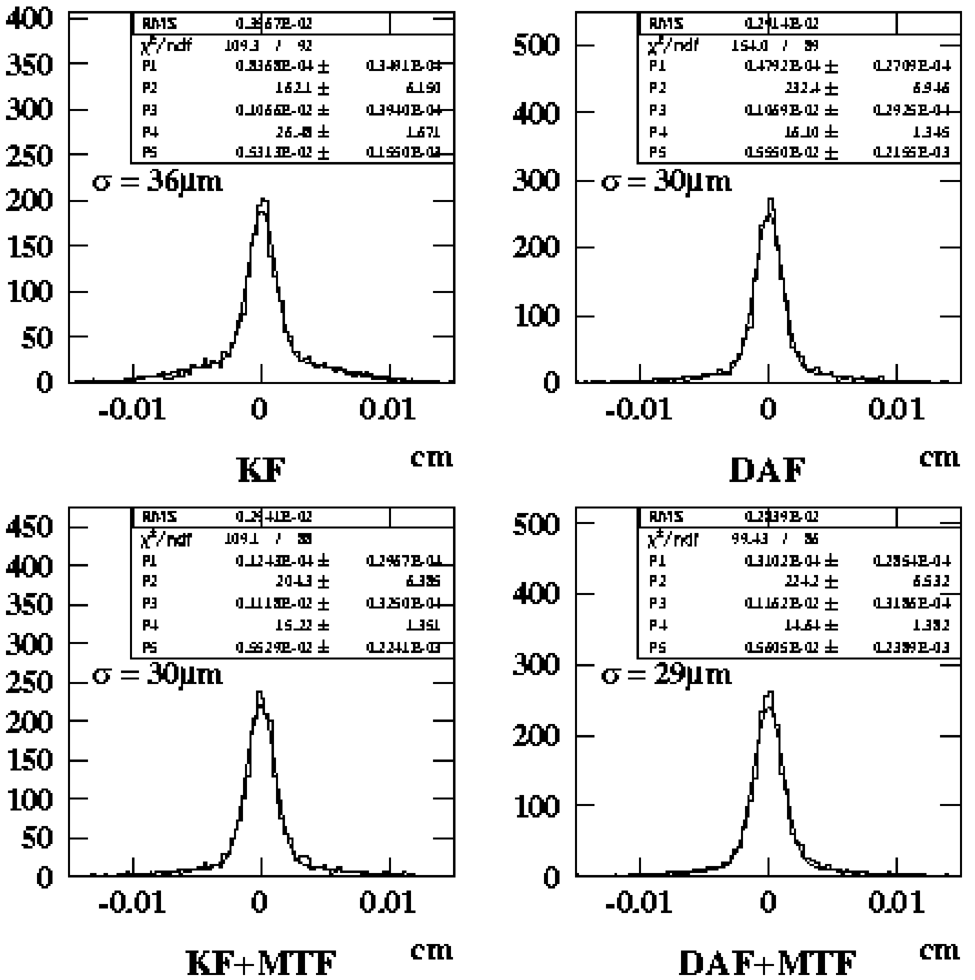
Adaptive filters: the DAF and the MTF

- Reconstruction of π tracks from the decay of high-p_T τ:
  - $H^0 \rightarrow \tau^+\tau^-$, $m(H^0) = 500\ \mathrm{GeV}/c^2$
- KF: Kalman Filter alone
- DAF: DAF with seed from KF
- KF+MTF: MTF tracks, seeded with KF tracks
- DAF+MTF: MTF tracks, seeded with DAF tracks

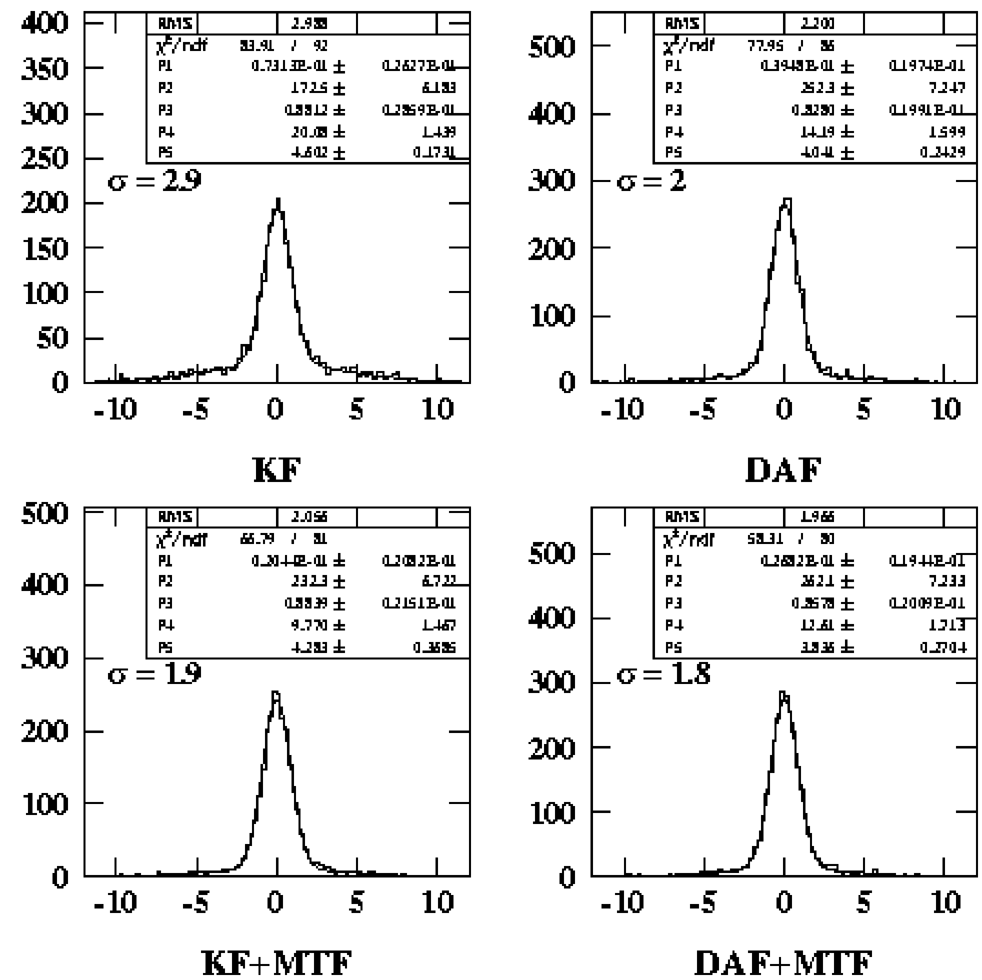


Adaptive filters: the DAF and the MTF

Transverse IP resolution



Transverse IP pull



Little improvement with the MTF over the DAF

Adaptive filters: the DAF and the MTF

- For "isolated tracks", even at high luminosity, the DAF and/or the MTF do not provide a measurable improvement in track quality
- DAF: "dense environment", e.g. b-jet with ET=200 GeV , τ jets:
  - Better track parameter resolutions and error estimates
  - Better track quality (χ^2)
- MTF: little improvement over DAF at the expected track densities!
  - little improvement on track parameter resolution
  - slightly better error estimate
  - slightly better overall track quality
- Better hit assignment (slightly lower fake rate)
- Seeding delicate (esp. MTF)
  - Better seeding methods would be needed
- Slower then standalone KF, use where appropriate

Conclusion

- A large number of algorithms have been implemented and are being evaluated
  - Several of these algorithms have never been tried before!
  - We have an array of tools to cope with different situations
- LSM shown to give very good (surprisingly good!) results even in difficult environments:
  - Well understood properties
  - Physicists know how to handle them and interpret the results.
- Combinatorial Kalman filter:
  - Efficient and robust pattern recognition
  - High efficiency, low fake rate
  - Low contamination from spurious hits, even with PU
  - Reconstruction ambiguities are solved after the first few layers
  - Good performance, suitable for high luminosity or heavy ion collisions
  - Fast enough for to be used in the HLT
  - Good track parameter resolutions after using only the first five to six hits
- Adaptive algorithms show improvements w.r.t. LSM in difficult situations

Backup

Alignment in CRAFT08

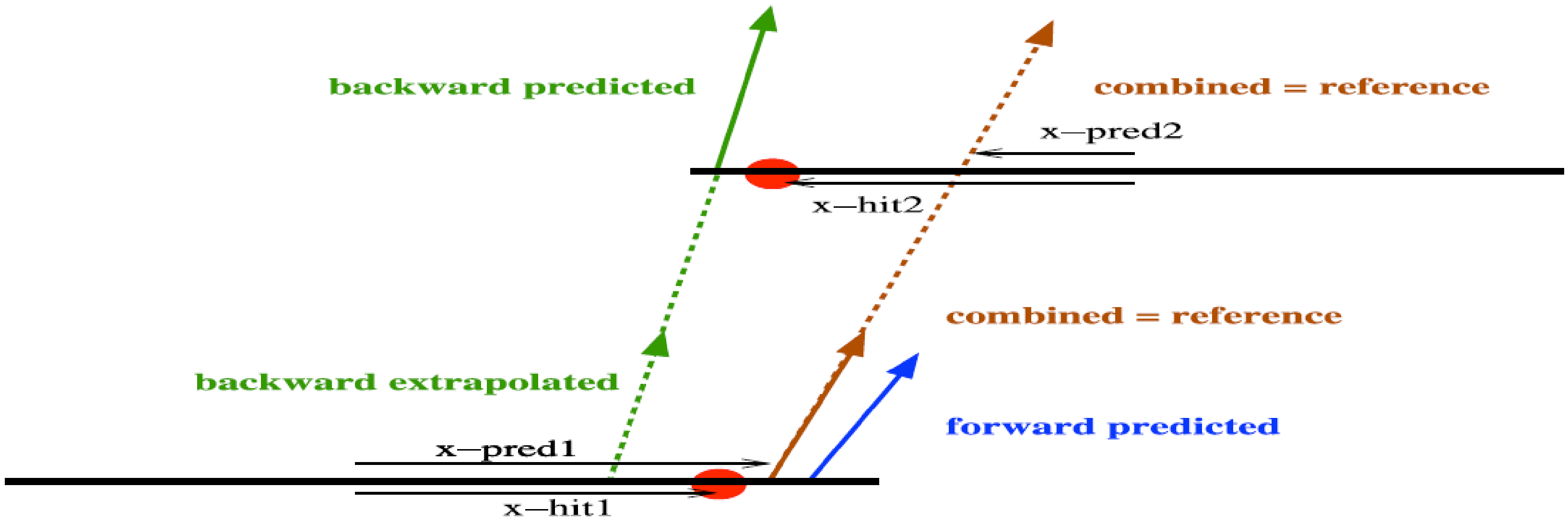
- Alignment strategy is based on running in chain two algorithms (16624 modules x 6 degrees of freedom):
  - local method on top of the geometry produced by global method
- Alignment with CRAFT08 better than expected

DMR RMS (µm)	Data Alignment	MC alignment	Ideal alignment	Modules >30 Hits
BPIX (x)	2.6	2.1	2.1	757/768
BPIX (y)	4.0	2.5	2.4	757/768
FPIX (x)	13.1	12.0	9.4	391/672
FPIX (y)	13.9	11.6	9.3	391/672
TIB (x)	2.5	1.2	1.1	2623/2724
TOB (x)	2.6	1.4	1.1	5129/5208
TID (x)	3.3	2.4	1.6	807/816
TEC (x)	7.4	4.6	2.5	6318/6400

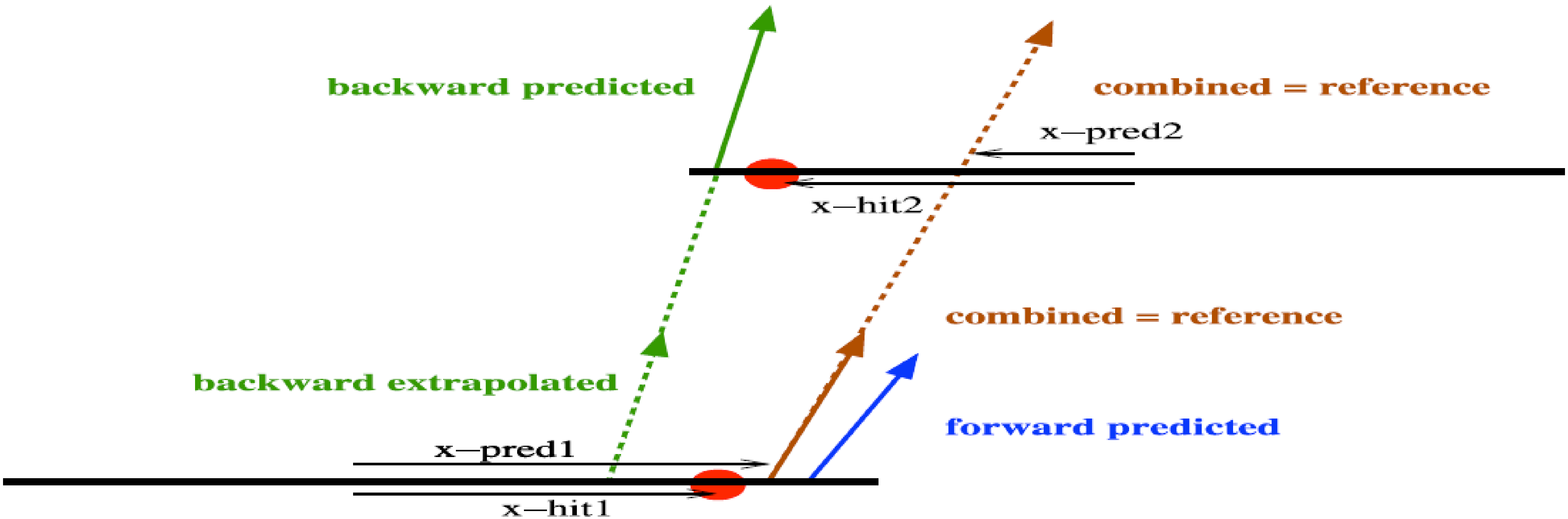
- Alignment validation
  - Track residuals and global parameters, sensor overlaps and geometry comparisons

Overlap study

- Use overlaps to measure hit resolution, validate and monitor alignment
  - "Overlap" = reconstructed track crossing a layer in the overlapping part of two adjacent modules
- Using overlaps:
    - Reduces the amount of material between two layers
    - Reduces the effects of track extrapolation
  - Compare the predicted position from the track fitting at each module in the overlap with the position of the hits
- Asses relative displacement and rotations between adjacent modules



Overlap study



- Combine "forward predicted" and "backward predicted" tracks to give best possible track fit without using the hits in the overlap pair
- A better precision can be obtained by using the difference between the predicted position $\Delta x_{pred} = x_{1pred} - x_{2pred}$, since it accounts for possible correlations between the modules
- Measure the accuracy of the prediction with the "double difference" $DD = \Delta x_{pred} - \Delta x_{hit}$ (with $\Delta x_{hit} = x_{1hit} - x_{2hit}$)

Track parameter resolution

Table 7: Standard deviation, mean, and 95% coverage of the residual and pull distributions of the track parameters. The units indicated pertain only to the residual distributions.

Track parameter	Residual distributions			Pull distributions		
	Std. Dev.	Mean	95% Cov.	Std. Dev.	Mean	95% Cov.
p_T (GeV/c)	0.083	0.000	1.92	0.99	0.01	2.1
Inverse p_T (GeV^{-1}c)	0.00035	0.00003	0.00213	0.99	-0.01	2.1
ϕ (mrad)	0.19	0.001	0.87	1.08	-0.02	2.4
θ (mrad)	0.40	0.003	1.11	0.93	-0.01	2.1
d_{xy} (μm)	22	0.30	61	1.22	0.00	2.9
d_z (μm)	39	0.28	94	0.94	-0.01	2.1